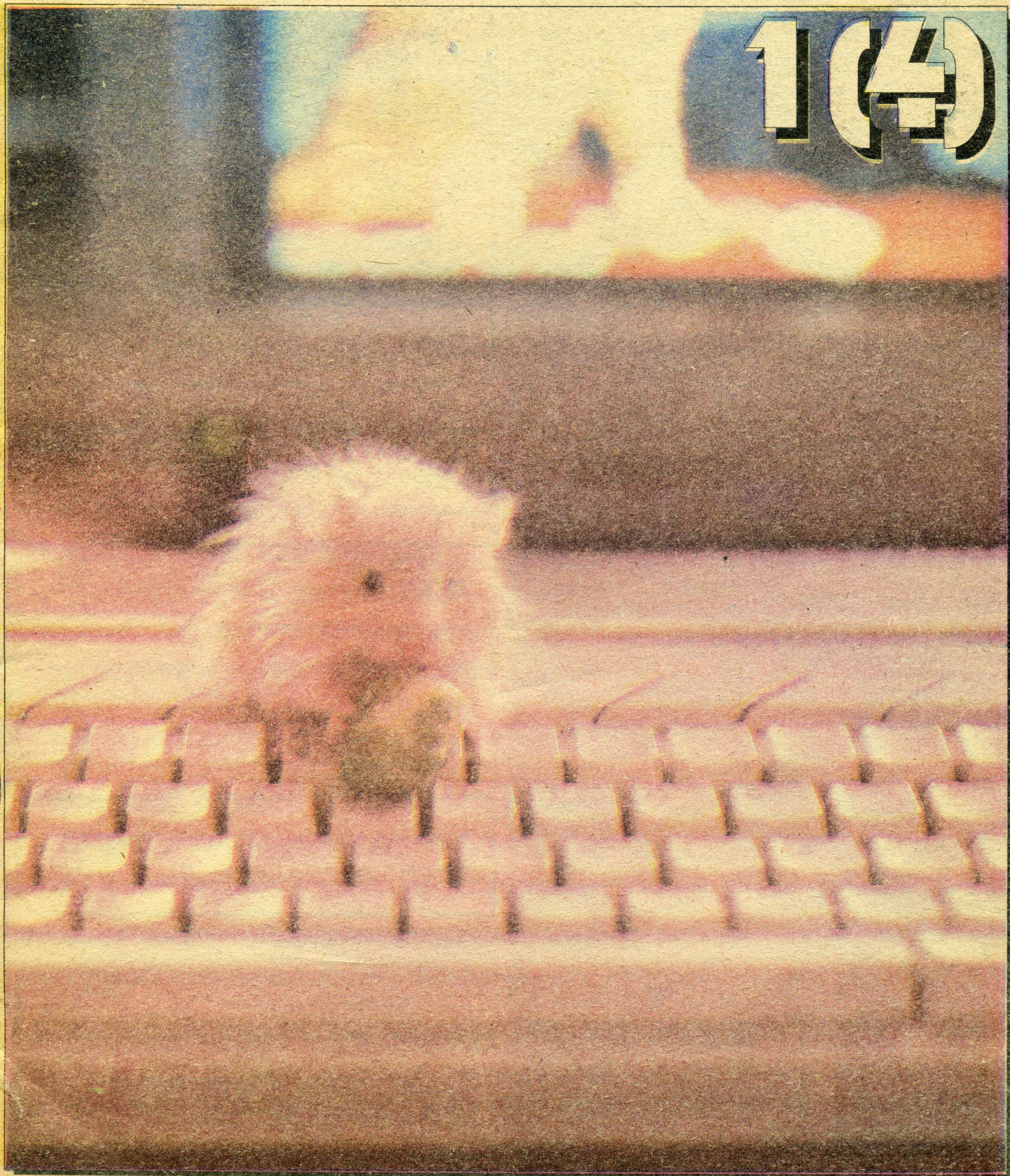


KOMMbiOTer

ISSN 0236-1949

1(4)





Уважаемый читатель!

Редакция "Компьютер" постоянно чувствует вашу моральную поддержку, выраженную в сотнях писем, полных симпатии, дружелюбия, искренней заинтересованности в успехе нашего дела.

"Я живу в Одессе, и хотя наш город довольно большой, но все же журналов, подобных вашему нет. (...) Обязательно буду покупать ваш журнал, за него и 1 р. 50 к. не жалко, честное слово" (А.Пузыренко, г.Одесса).

"Наконец-то и советские пользователи стали получать именно то, что им в принципе нужно" (С.Г.Зубарев, г.Москва).

"Очень нужное дело Вы начали: информация, публикуемая на страницах "Компьютера", нужна не только пользователям ПК, но и основной массе учащейся молодежи" (О.Э. Анненков, г. Иркутск).

Иногда у нас даже начинает кружиться голова от собственных заслуг:

"Уважаемая редакция, глубоко признателен Вам за то, что Вы первые в СССР упомянули о существовании в мире таких компьютеров, как Atari ST, Commodore 500/2000, Macintosh. К великому сожалению, наша страна опять повторяет ошибки 60-70 годов, ориентируясь только на компьютеры фирмы IBM (хотя я и не могу умалять их достоинства)" (Е.В.Силантьев, г.Ростов-на-Дону).

Многие рассказывают о своих трудностях, и это дает нам представление о проблемах общенациональных:

"К счастью, я имею дома компьютер Atari 1040 ST, к несчастью, я живу в СССР, где нет вообще ничего по этим компьютерам и все мы варимся в собственных домыслах и слухах, доходящих из-за рубежа, не говоря уже о программах, которые все 3-4-летней давности" (С.Е.В.).

"Дома у меня нет компьютера. Родителям не до того, и я их уже замучил своими разговорами о компьютерах (...) Я понимаю, что Вы ничем не можете помочь, поэтому прощаюсь" (Самойлов Андрей, 13 лет, г.Москва).

В доброй половине писем содержится универсальный рефрен, с точностью до нескольких слов близкий к следующему высказыванию читателя:

"Журнал очень понравился своим разнообразием и популярностью. Было бы отлично, если будет на него подписка или что-то вроде индивидуальной подписки, или еще как-нибудь, потому что до таких городов как (...) этот журнал не доходит и может случиться, что Ваш первый номер был для меня и последним".

Во избежание значительного повышения цены редакция решила воздержаться от организации подписки сборника на 1991 г. Рекомендуем нашим читателям направлять заявки о высылке журнала наложенным платежом - с указанием необходимого количества экземпляров и начальным номером выпуска - по одному из адресов магазинов, указанных на с. 62 в этом номере. Мы сделаем все от нас зависящее, чтобы ранее присланные по адресу издательства заявки о приобретении сборника также были удовлетворены.

Мы благодарим всех читателей, чьи письма, содержащие дельную, конструктивную критику, ценные пожелания и советы, помогли нам определить тематику четырех номеров 1991 г., с учетом ваших интересов, потребностей, вкусов.

Огромное количество пожеланий относится к необходимости уделять больше места разделу "Компьютер дома" и, в его рамках - компьютерам ZX Spectrum, Atari и другим, поскольку имеется острый дефицит информации и программ для них. С.Михалюк из г.Ленинграда даже считает, что журнал "Компьютер" может стать своего рода клубом пользователей ZX Spectrum.

При традиционном разнообразии материалов и соразмерности разделов тематика номеров 1991 г. спроектирована следующим образом.

Настоящий выпуск (N 1) посвящен преимущественно домашним компьютерам, главным образом ZX Spectrum.

Доминантой второго номера журнала будут материалы по компьютерной графике, анимации, САПР.

Главной темой третьего выпуска будут гуманитарные применения компьютеров - в системе образования, медицине, экономике, политике.

Четвертый номер журнала будет посвящен преимущественно деловым и научным применениям ПК как домашних, так и профессиональных моделей.

С сожалением сообщаем вам, что в 1991 г. из-за удорожания бумаги и полиграфических услуг журнал будет стоить 2 руб. 50 коп., причем это повышение цены следует понимать не как отказ от нашей концепции издавать массовый и доступный большинству журнал, а как условие нашей финансовой выживаемости.

И последняя новость: в 1991 г., начиная с первого номера, наш журнал по примеру киевских и новосибирских коллег будет распространять электронное приложение (ЭП) - специальным образом сверстаный для экрана IBM-совместимых моделей ПК - текст с графическими и музыкальными заставками и развитым справочным аппаратом - перекрестными ссылками, указателями. Помимо собственной электронной версии соответствующих номеров сборника "Компьютер", программ распаковки архива и прокрутки файлов специального формата в комплект ЭП войдут поставляемые на отдельной дискете файл-"корзина" рекламно-коммерческих предложений подписчиков ЭП и файл-форма для внесения структурированных заявок подписчиков в следующий номер ЭП. Указанная дискета с заполненной заявкой может быть возвращена по желанию подписчика в редакцию сборника "Компьютер" для включения в "корзину" следующего выпуска ЭП. Цены на электронную версию журнала будут дифференцированы для индивидуальных подписчиков и организаций и останутся в рамках понятия "разумной достаточности". Приглашаем заинтересованные предприятия и организации направлять письменные заявки на подписку электронного приложения по адресу редакции (с пометкой на конверте - "ЭП").

Текстовые файлы выпущенных статей по-прежнему будут распространяться бесплатно через московский узел сети FIDO и редакцию журнала наряду с комплектом антивирусных программ.

Редакция воспринимает как новое доброе пожелание журналу следующие высказывания наших читателей:

"Очень хотелось бы, чтобы "Компьютер" стал массовым изданием, оперативным и авторитетным" (А.Л.Волотов, г.Вобруйск).

"Желаю Вам больших успехов, постоянного увеличения тиража (да минуя Вас сложности с бумагой) и возможного перехода на 6 или 12 номеров в году. Удачи Вам!" (А.Лемкиш, г.Запорожье).

В свою очередь мы от души поздравляем всех наших читателей с Новым годом. Желаем вам и вашим близким здоровья, счастья, удачи и успехов. Мы искренне верим, что в непростых для страны обстоятельствах у нас достанет мудрости и выдержки повернуть дела к лучшему.

Константин Коробов

Константин Коробов

Москва, Варшава, декабрь 1990 г.

Что, когда, где**МАРТ****CEBIT '91**

World Center for Office, Inf.
and Telecommunications
Technology

21-28 марта

Ганновер, ФРГ

Организатор: Deutsche
Messe AG Messagelände,
D-3000 Hannover 82, BRD

телефон: 0511-890

факс: 0511-8932626

АПРЕЛЬ**COMTEK**

Международная компью-
терная выставка

8-12 апреля

ВДНХ Москва

Организатор: Crocus Interna-
tional, ул. Миклухо-Маклая,
37а, 117483 Москва, СССР

телефоны 335-1444, 330-3219

факс (095) 420-2265

INFOSYSTEM

Ежегодная международная
компьютерная ярмарка

апрель 1991

Познань, Польша

Организатор: Международ-
ная познаньская ярмарка,
ул. Плоговска, 14, 60-734

Познань, Польша

телефон: (081) 699-341

телекс: 413251 targ pl

MIPEL '91

Международная выставка
промышленной электро-
ники и электротехники

9-12 апреля

Будапешт, Венгрия

Организатор: HUNGEXPO,
P.O. box 44 Budapest,
Hungary

телефон 361-573555

телекс: 224684 hexpo h

МАЙ**СВЯЗЬ '91**

22-31 мая

Москва, Красная Пресня

Организатор: Экспосентер, ул.
Сокольнический вал, 1, 107113
Москва, СССР

телекс: 411185 expro su

факс 288-95-37

ПОЛИГРАФБУММАШ

Международная выставка
печатного дела, в том числе
настольные издательские
системы

16-23 мая

Москва, Сокольники

Организатор: Glahe Interna-
tional KG, Herler Strasse
103-109, D-50000 Koeln 80, BRD

телефон: 0221/694011

телекс: 8 874 676

факс: 0221/695865

CAT '91

Computer Aided Techno-
logies Manufacturing

14-17 мая

Штуттгарт, Германия

Организатор: Stuttgarter
Messe- und Kongress GmbH,
AM Kochenhof 16, 7000
Stuttgart 10, BRD

телефон: 0711/25890

телекс: 722584 killb d

факс: 0711/2589440

**MICROELETTRONICA
& TECNIA**

Выставка промышленных
применений электроники и
информатики

9-12 мая

Виченца, Италия

Организатор: Ente Fiera di
Vicenza, via dell'Oreficeria,
36100 Vicenza, Italy

телефон: 04444/969111

телекс: 481542 tiervi

факс: 0444/563954

SEMICON WEST

Выставка полупроводников

21-23 мая

Сан Матео, США

Организатор: Semiconductor
Equipment & Material
International, 805 E
Middlefield Rd., Sunnyvale,
CA 94043, USA

телефон: 415/9645111

телекс: 856777

факс: 415/9675375

MIKRODATA

Выставка применений ЭВМ
дома, в офисе и в
процессе обучения

22-25 мая

Осло, Норвегия

Организатор: Norges
Varemesse, P.O. box 130,
Skoyen, 0212 Oslo 2, Norway

телефон: 02/438080

телекс: 78748 messe n

факс: 02/431914

EUROTRONIC

Европейская компьютерная
ярмарка

27-31 мая

Брюссель, Бельгия

Организатор: Brussels Int'l
Trade Fair, Place de Belgique,
1020 Brussels, Belgium

телефон: 02/4770477

факс: 02/4788023

**LOGIC-COMPUTER
SHOW**

Международная
компьютерная ярмарка

28 мая - 1 июня

Цюрих, Швейцария

Организатор: Int'l Trade
Fairs & Special Exh.,
Thurgauerstrasse 7, 8050
Zurich, Switzerland

телефон: 01/3115055

факс: 01/3119749

ИЮНЬ**INFORMATION
TECHNOLOGY '91**

Ежегодная Международная
компьютерная выставка

29 мая - 5 июня

Минск, Белорусская ССР

Организатор: MinskExpo,
проспект Машерова, 14,
Минск, БССР

телефон: 0172/269014;
095/2079667

телекс: 252190 makra

факс: 0172/269936

ТЕХНОЛОГИЯ '91

Международная компью-
терная и промышленная
ярмарка

10-16 июня

Пермь, РСФСР

Организатор: Glahe
International KG, Herler
Strasse 103-109, D-50000 Koeln
80, BRD

телефон: 0221/694011

телекс: 8874676

факс: 0221/695865

PC EXPO NEW YORK

Международная выставка
персональных ЭВМ и
периферии

5-7 июня

Нью Йорк, США

Организатор: H A Bruno Inc.
385 Sylvan Ave., Englewood
Cliffs, NJ 07632, USA

телефон: 201/5698542

факс: 201/5691153

BUSINESS EXPO

Выставка деловых
применений ЭВМ

12-14 июня

Крайстчерч, Новая
Зеландия

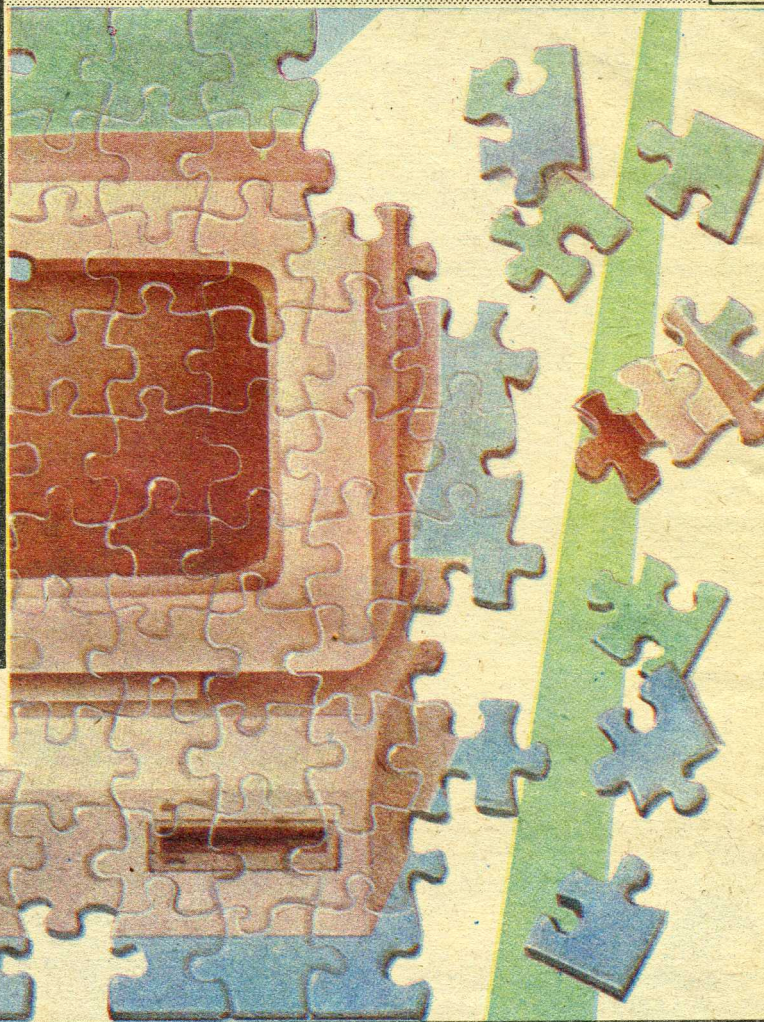
Организатор: XPO EXH Ltd., 5
Cheshire Str., Parnell,
Auckland, New Zealand

телефон: 09/397990

факс: 09/793358

В мире "Компьютера"

-  Сэр Клайв Синклер
-  Компьютер 1990 года
-  SMAU '90
-  Что такое компьютерный клуб
-  Болгарский клон на советском рынке
-  Какие "Спектрумы" ходят в Союзе
-  Средства создания экспертных систем



В мире "Компьютера"

© Халина Мадейчик

Сэр Клайв Синклер

Если Стивен Джобс по праву считается отцом персонального компьютера, то как назвать человека, который преподнес компьютер в подарок домохозяйкам, а большинству наших читателей открыл дорогу в новый, таинственный мир байтов, файлов, Бейсика?

Клайв Синклер родился в 1940 г. В 22 года, основав свою первую фирму Sinclair Radionics, он стал продавать разработанные им комплекты для сборки миниатюрных радиоприемников. В 1972 г. он изготовил карманный калькулятор, который назывался Executive и стоил "всего" 100 дол. В 1977 г., опередив японцев, К. Синклер разработал и выпустил первый карманный телевизор - Microvision, стоимостью 300 дол. Не прошло двух лет, и цену пользовавшегося огромным спросом "телика" можно было понизить напо-

лону. Деньги, полученные от продажи своих изобретений Синклер решил "вложить" в новую фирму - Sinclair Research. Она была зарегистрирована в июне 1979 г. 10 месяцев спустя на рынке появился ZX80. Благодаря весьма демократичной цене - всего 100 фунтов стерлингов - он стал пользоваться большим спросом, но не в Великобритании, а на континенте, куда направлялась большая часть продукции. Через полгода во Франции был продан стотысячный ZX80.

Однако Синклер явно не хотел почитать на лаврах. Его ум, захваченный идеей массовой компьютеризации, продолжал работать. В марте 1981 г. началось производство ZX81 - первого подлинно домашнего компьютера. Благодаря цене (100 дол.) он был доступен всем желающим. Подключался к бытовому телевизору, накопителем служила магнитофонная лента, а время, проведенное за заполнением его ОЗУ, насчитывавшего всего 1 Кбайт, я буду долго вспоминать. Именно тогда я усвоила основные приемы программирования на Бейсике (программ для ZX81 тогда еще не было). За 2 года таких энтузиастов компьютерного дела, как я, набралось больше миллиона.

В июне 1982 г. началась эра Спектрумов. Первая модель - 48-Кбайтовый "резиновый" ZX Spectrum - стоила 175 фунтов стерлингов. Год спустя он обошелся моей семье всего в 150 дол.

За свои разработки и славу, которую он принес английской короне, Синклер был удостоен дворянского титула. С 1981 г. перед его фамилией стали появляться три маленькие буквы - сэр.

Ум гения не укладывается в привычные понятия. И в 1984 г. Синклер разработал и выпустил на рынок совершен-

но новый компьютер - QL, который до сих пор, по мнению многих специалистов, остается лучшим домашним компьютером для профессиональных применений. Однако эра домашних компьютеров уже подходила к концу. Беспрепятственно посматривавшая на Америку Европа дождалась, наконец, разработки, моментально ставшей стандартом. Началась эра IBM.

Хотя Клайв Синклер и продолжал выпускать новые версии Спектрумов (Plus и 128), но на самом деле он был занят совершенно другой идеей - разработкой дешевой электромашины. В 1985 г. фирма Sinclair Research стала ее выпускать под названием C5. Но на этот раз удача покинула гения. Деньги, потраченные на разработку C5, не принесли ожидаемого дохода. Через год права на все продукты Sinclair Research пришлось продать фирме Amstrad. Это были "похороны" Спектрума. Алан Шугер - владелец Amstrad - выпустил только одну модель - Спектрум 128 со встроенным 3-дюймовым дисководом.

Сам Синклер продолжает свои разработки в новой фирме - Cambridge Computers. В 1987 г. появился Z88 - портативный IBM-совместимый компьютер, в прошлом - laptop Psion MC400, в котором управление курсором осуществляется не мышью, а передвижением пальца по специальной сенсорной табличке. Продолжается разработка новой, MS-DOS-совместимой машины с операционной системой, "прошитой" в ПЗУ.

В то же время Синклер не забывает о "грехах" своей молодости. В 1989 г. он выпустил на рынок недорогой приемник спутникового телевидения. Достоин уважения и его настойчивость, поскольку, несмотря на неудачу, он продолжает разрабатывать в Sinclair Research новую модель недорогой городской электромашины C15. Но на этот

раз сэр Клайв решил подстраховаться и параллельно с C15 готовит к выпуску "сверхпортативную", складную, почти карманную модель велосипеда. Как далеко "заедет" на нем в будущее Клайв Синклер - покажет время.

Июль 1979 - основание фирмы Sinclair Research

Февраль 1980 - на рынке появился ZX80

Март 1981 - начало продажи ZX81

Июнь 1982 - появление ZX Spectrum (48 Кбайт ОЗУ)

Апрель 1984 - продажа первых экземпляров QL

Январь 1985 - начало производства ZX Spectrum Plus и Spectrum 128

Апрель 1986 - право на выпуск всех компьютеров Sinclair Research продано фирме Amstrad

Август 1986 - начало производства Spectrum +2 со встроенным магнитофоном

Февраль 1987 - Amstrad начинает выпуск Spectrum +3 со встроенным 3-дюймовым дисководом

Март 1987 - новое поколение синклеровских компьютеров открывает Z88

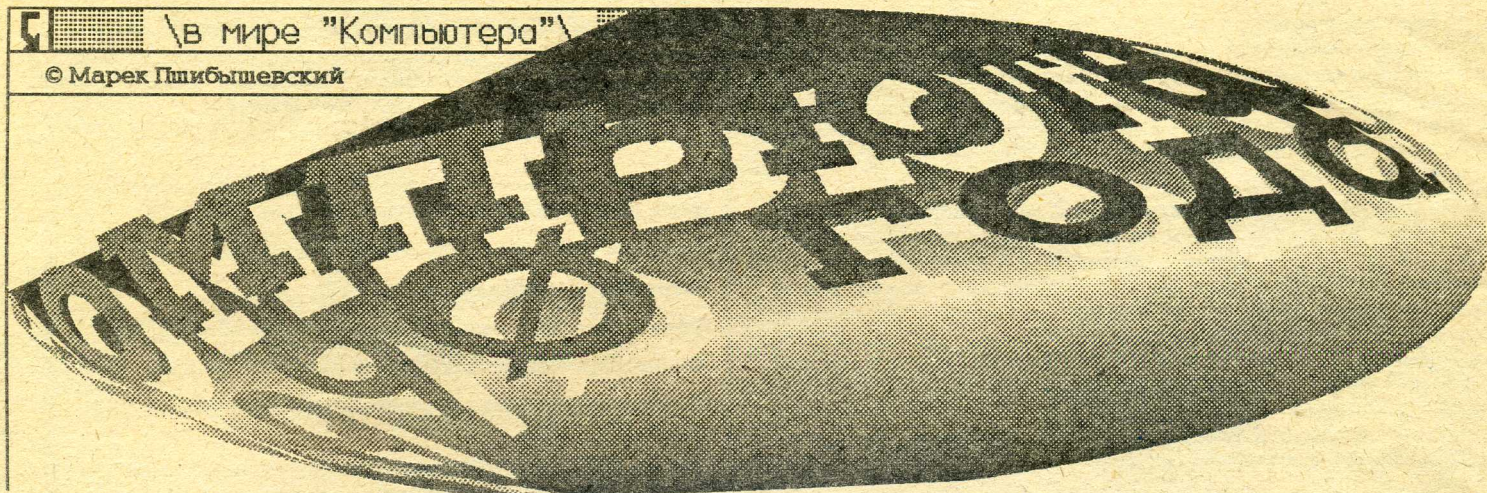
Октябрь 1989 - начало производства Psion MC400 (лучший laptop 1990 года в анкете журнала "CHIP")

Август 1990 - фирма Amstrad принимает решение о прекращении производства Spectrum +3




 \ в мире "Компьютера" \

© Марек Пшибышевский



Каждый год журнал "СНIP" из ФРГ проводит опрос компьютерной прессы разных стран, цель которого - присуждение звания лучшей ЭВМ года. В отдельных категориях ЭВМ в 1990 г. лидерами были признаны следующие компьютеры.

Домашние компьютеры

Второй раз подряд журналисты из Европы, США и Японии признали лучшим в этой категории компьютер фирмы COMMODORE - Amiga 500 (170 очков).



Лишь на 25 очков меньше набрал его главный конкурент - Atari 1040 ST. Большой неожиданностью для многих стало 3-е место, занятое "старичком", 8-разрядным компьютером CPC 6128 британской фирмы AMSTRAD, выпускаемым с 1985 года.

Вместо описания старого компьютера Amiga 500 я предлагаю читателям информацию о его младшем брате - Amiga 3000, поскольку именно этот компьютер, появившийся в продаже в сентябре прошлого года, открывает фирме COMMODORE дверь на рынок профессиональных ЭВМ.

Amiga 3000 предназначен для применения в новой области передачи информации, которая называется multimedia (объединяющей компьютерную графику, музыку, синтез речи, видео). В сером корпусе находится главная плата с 32-разрядным процессором Motorola 68030 (тактовая частота в зависимости от варианта - 16 или 25 МГц), ОЗУ емкостью от 2 до 18 Мбайт, дисковод 3,5-дюймовых ди-

сков 880 Кбайт, жесткий диск SCSI 40 или 100 Мбайт (время доступа - 19 миллисекунд). С правой стороны выведены разъемы для клавиатуры, мыши и джойстика. Сзади - полный комплект интерфейсов: RS 232, Centronics, SCSI, audio stereo, Amiga Video, VGA, сети Ethernet и дополнительного внешнего дисковод.

Новая операционная система Kickstart 2, которая заменила Amiga-DOS, "прошита" в 512 Кбайтах ПЗУ. Что интересно - пользователь может выбрать язык, на котором будет проходить общение с компьютером (английский, немецкий, французский).

Портативные ЭВМ типа "laptop"

В этой категории лучшим признан компьютер MC400 британской фирмы PSION.



Весящая всего 2,3 кг "малышка" (не совместимая с MS-DOS) отличается оригинальностью решения проблемы управления курсором на экране. Над клавиатурой размещена небольшая сенсорная табличка. Курсором пользователь управляет, водя пальцем по этой табличке.

Второе место заняла японская портативная ПЭВМ Sharp 6220, а третье - описанная в первом прошлогоднем выпуске "Компьютера" модель Portfolio фирмы ATARI.

Сердцем MC400 является процессор Intel 80C86. 256 Кбайт ОЗУ дополняет 526 Кбайт ПЗУ с "прошитой" операционной системой, текстовым процессором, небольшой базой данных и электронной записной книжкой. Общение с внешним миром - через Centronics или RS-232C. Данные могут храниться на специальных картах памяти (Flash-EPROM) емкостью 512 Кбайт.

IBM PC/XT-совместимые ПЭВМ

И в этой категории победителем стала также европейская фирма SCHNEIDER, представившая компьютер EURO PC II. Второе место занял почти карманный Сагу 1 тайваньской фирмы FLUTECN, третье - голландская ПЭВМ PC Compact 2 фирмы TULIP.

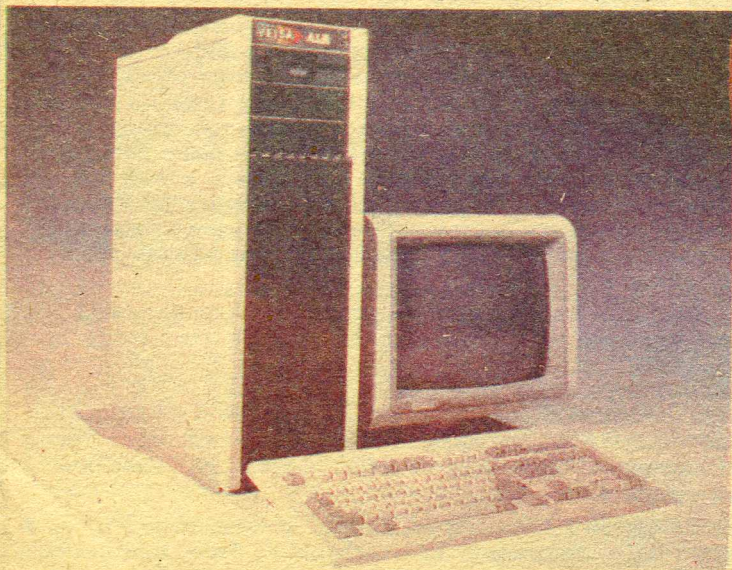
Отличающийся изысканностью дизайна EURO PC II - редкая на рынке модель XT, в которой процессор, дисковод и клавиатура объединены в одном корпусе.



Микропроцессор Intel 8088 работает с тактовой частотой 9,54 МГц. Машина имеет 768 Кбайт ОЗУ, дисковод 3,5-дюймовых дисков, возможность подключить любой 3,5- или 5,25-дюймовый жесткий диск. Машина поставляется с MS-DOS 3.3 и пакетом Works.

ПЭВМ с процессором 80386 и i486

О первых ПЭВМ с микропроцессором Intel i486 мы уже писали. За это время они стали настолько популярными, что журнал "CHIP" решил ввести в свою анкету новую категорию машин. Победителем стал компьютер PowerVEISA американской фирмы Advanced Logic Research (ALR).



В машине устранены все недостатки, упоминавшиеся в первом прошлогоднем выпуске нашего журнала. Она поставляется сегодня с микропроцессором Intel 80386 или i486 (по выбору) и сопроцессором 80387. ОЗУ с начальным объемом 5 Мбайт можно расширить до 17 Мбайт. Имеется дополнительная плата с 49 Мбайтами ОЗУ. Кэш-буфер насчитывает 64 Кбайта, жесткий диск - 80 Мбайт. Его контроллер установлен на главной плате, а не в виде привычной отдельной карты. Два стандартных НГМД (3,5- и 5,25-дюймовых). В компьютере шесть системных шин: три 32-битовые, соответствующие стандарту EISA, две 16-битовые шины AT и одна 8-битовая XT. Пользователь может заказывать машину с одной из двух операционных систем - MS-DOS или Unix.

ПЭВМ с процессором Motorola 680x0

В этой категории борьба была самой напряженной. Победитель - Macintosh IIx американской фирмы APPLE (120 очков) - опередил всего на 10 очков знаменитый NeXT (мы писали о нем в N 1 сборника "Компьютер" за 1990 г.) и на 15 очков - машину Джека Трэмбела Atari Mega ST (на которой верстается наш журнал).



В компьютере Macintosh IIx микропроцессор Motorola 68030 с тактовой частотой 40 МГц (!) работает в паре с сопроцессором 68882. 4 Мбайта ОЗУ можно расширить до 32 Мбайт. Macintosh IIx снабжен 32-Кбайтовым кэш-буфером и имеет 512 Кбайт ПЗУ, стандартный для компьютеров этой фирмы дисковод для 3,5-дюймовых дискет емкостью 1,44 Мбайта, жесткий диск 80 или 160 Мбайт.

В завершение несколько слов о машине Стива Джобса, которая постепенно начинает обретать все большую популярность как workstation. В конце прошлого года Джобс решил ускорить этот процесс и выпустил три новые модели NeXT. Новинкой является встроенный дисковод для 3,5-дюймовых дискет емкостью 2,44 Мбайта, который читает файлы формата MS-DOS. Самая дешевая из новых моделей стоит всего 5000 дол. "Всего", поскольку это весьма умеренная цена за машину с процессором Motorola 68040 с тактовой частотой 25 МГц, 8 Мбайтами ОЗУ, 100-Мбайтовым жестким диском, к которой можно еще подключить 256 Мбайтовый оптический диск. Станет ли NeXT компьютером 1991 г. - покажет время.

Перевод Анджее Поплавского

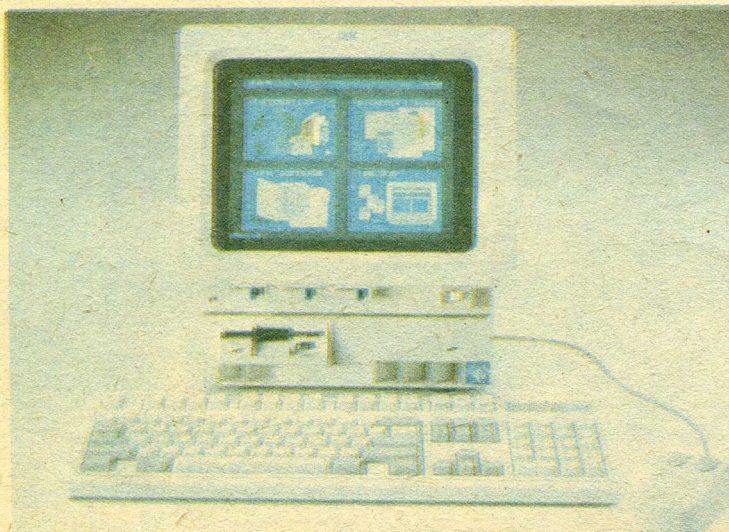




SMAU - это вторая по величине и значению европейская ярмарка средств информатики и связи (первое место занимает ганноверский CeBIT), которая ежегодно проводится в Милане. В октябре 1990 г. она проходила уже в 23-й раз. 961 фирма демонстрировала свои продукты на 148 тыс. кв. м. Для сравнения: в крупнейшей из состоявшихся до сих пор в СССР компьютерной выставке ("Информатика-90", октябрь 1990 г., Москва, Краснопресненский выставочный комплекс) участвовало 360 фирм, а выставочная площадь незначительно превысила 10 тыс. кв. м.

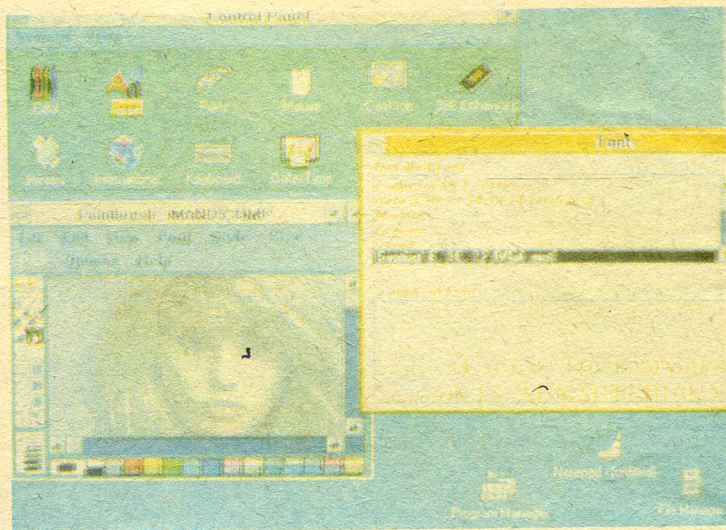
"Шлягерами" SMAU были система PS/1 фирмы IBM, которая демонстрировалась как домашний компьютер нового поколения (в том числе игровой, именно у стенда IBM было больше всего молодежи), и широко рекламируемая фирмой MICROSOFT программная система Windows 3.0. Стоит отметить, что количество прикладных программ, работающих в среде Windows, растет буквально изо дня в день.

Семейство PS/1, представленное четырьмя моделями, основано на микропроцессоре Intel 80286 с тактовой частотой 10 МГц. Компьютеры этой линии снабжены видеоадаптером VGA, дисководом 3,5 дюйма и модемом с декодером видеотекста. В ПЗУ "прошита" ОС MS-DOS 4.01. К каждому проданному экземпляру прилагаются (бесплатно!) пакет MS Works и абонемент видеотекста на 3 месяца.



Самый дешевый компьютер из серии PS/1 стоит 999 дол.: монитор VGA моно, ОЗУ 512 Кбайт, 1 дисковод 3,5 дюйма. Более требовательным покупателям за 1999 дол. предлагается модель с винчестером 30 Мбайт, цветным VGA-монитором, ОЗУ 1 Мбайт. Самый дорогой вариант стоит 2700 дол.

Что касается Windows 3.0, то этот пакет имеет большие шансы стать основным стандартом на рынке программ-оболочек операционных систем для IBM-совместимых машин и опередить своего главного конкурента - GEM фирмы DIGITAL RESEARCH. В Милане пакет Windows 3.0 завоевал главный приз конкурса промышленного дизайна в области программного обеспечения.



DOS, судя по всему, так и останется незаменимой, тем более, что фирма MICROSOFT в конце прошлого года выпустила новую, уже пятую версию своей операционной системы MS-DOS - более "дружественную", с улучшенной оболочкой и новыми командами.

Были на SMAU и другие новинки. Фирма AMSTRAD из Великобритании показала новую серию своих персональных компьютеров с обозначением 3000, французская фирма GOUPIIL демонстрировала ПЭВМ с процессором 486 и архитектурой, основанной на стандарте EISA, а известный своими принтерами STAR привез в Милан три новых печатающих устройства: LC-20 и LC-200, которые в ближайшее время займут место LC-10 и LC-10 Color, а также LC24-200 - цветную версию 24-игольного принтера LC24-10.

Завсегдатаев компьютерных ярмарок и выставок разочаровало отсутствие возможности ознакомиться с "золотой" Вентурой, о которой мы писали год назад. Представитель фирмы RANK XEROX сообщил, что вопреки сообщениям печати о начале продажи новой, работающей в среде Windows версии, к миланской ярмарке ее еще не успели подготовить. На стендах появился Clipper 5.0 фирмы NANTUCKET. Продажа этой программы в США была приостановлена вскоре после "премьеры", а ее русская версия была разработана почти одновременно с основной, англоязычной версией. Autodesk демонстрировал в Милане 11-ю версию системы AutoCAD.

В этом году ярмарка SMAU проходила под девизом "наведение мостов". Имелось в виду налаживание контактов между Востоком и Западом. Этим объясняется присутствие на сопутствовавшем ярмарке семинаре "К востоку от SMAU" мэра Москвы Гавриила Попова, министра информатики России Владимира Булгака, директора Института проблем информатики АН СССР Игоря Мизина, других представителей СССР, министров и замминистров из Чехо-Словакии и Венгрии, а также представителей, связанных с информатикой, из Польши и Югославии. На мой вопрос о причине интереса к Востоку директор ярмарки Эноре Деотто (Enore Deotto) ответил: "Когда была разрушена берлинская стена, мы увидели совершенно новый для нас мир, огромный рынок, восхитительную культуру. Нас интересуют события в Восточной Европе. По мере своих скромных возможностей мы хотим помочь итальянцам узнать о вас как можно больше".

Востоком Европы интересуются также французы, а подтверждением тому служит факт, что они готовятся к проведению в этом году в Москве восточно-европейского варианта парижской компьютерной ярмарки SICOB, третьей по значению в Европе. В проведении выставки в Москве заинтересована также дирекция SMAU, которая ведет с представителями советских выставочных фирм интенсивные переговоры.

В беседе с директором Деотто я предложил, чтобы к участию в следующей ярмарке SMAU были приглашены представители компьютерной прессы из СССР и Польши.

Предложение было принято с большим интересом. Для большинства итальянцев это была бы первая возможность получить информацию о компьютерном рынке в наших странах. Нам, журналистам, это дало бы шанс установить более тесные контакты с мировой компьютерной прессой, которая, отметим, в Милане имела собственный павильон, где демонстрировала свои издательские достижения.



На каждой выставке (на ярмарку SMAU меня пригласила польская фирма PRO-INFO), я стараюсь найти нечто курьезное, что могло бы представлять собой интерес для читателей. В Милане мне попался комплект из 6 дисков CD-ROM (compact disc read only memory), а записана на них была... история эротической фотографии. 6000 фотографий периода 1839-1930 гг. являются якобы лишь введением в дальнейшие исследования влияния эротики на всевозможные области жизни. Исследования ведет миланская фирма SCLA PUBLISHING GROUP. Ну что ж, в стране, где порно-звезда Чиччолина заседает в парламенте, это, наверное, никого не может удивить.

Перевод Анджея Поплавского

В мире "Компьютера"

© Владимир Федоров

Что такое компьютерный клуб

Для эпохи персональных компьютеров характерен новый тип независимых общественных организаций - компьютерные клубы, или ассоциации пользователей. Их число давно перевалило за тысячу. Авторитет ассоциаций очень высок. Крупнейшие корпорации вкладывают большие средства для поддержки этого движения, прислушиваются к мнению его членов, готовят специальные презентации для конференций пользователей. Это и неудивительно - ассоциации обеспечивают обратную связь "пользователь-производитель", проводят тестирование и оценку различных продуктов, от них зависит успех того или иного продукта на рынке.

6 октября 1990 г. во Франкфурте проходила ежегодная конференция европейской Ассоциации пользователей (AUGE). Руководитель секции Apple II Мартин Георг (Martin George) рассказал о деятельности этого старейшего и крупнейшего в Германии компьютерного клуба. В настоящее время клуб насчитывает около 1800 членов. Годовой взнос составляет 100 марок, студентам предоставляется скидка в 30 марок. AUGE объединяет пользователей различных компьютеров: Apple II, Apple Macintosh, IBM PC, Atari ST, Commodore Amiga. Распределение пользователей по операционным системам таково (в процентах): Apple Dos - 23, ProDos и GS/OS - 56, MS-DOS и OS/2 - 30, Mac OS - 12, Atari TOS - 11, CP/M - 14, UNIX - 7, Amiga OS - 2.

Члены AUGE регулярно встречаются более чем в 30 региональных группах, расположенных по всей стране. На эти встречи приходят пользователи самых разных компьютеров. Люди собираются для того, чтобы увидеть новые программы, встретиться с разработчиками, обменяться новостями, поговорить в непринужденной обстановке, скопировать некоммерческое (PD - Public Domain) программное обеспечение.

Основная деятельность AUGE ведется через клубы по интересам - так называемые SIGs (Special Interest Groups). Эти группы выпускают собственные журналы (как правило 4-6 номеров в год) и проводят несколько встреч в год. Крупнейшая группа в AUGE объединяет 280 пользователей ПК Apple II GS. Другие группы специализируются на применении компьютеров для бирж, в области банковского дела, телекоммуникаций, радиосвязи. Есть группы по ПК Macintosh, Amiga, Atari ST, операционным системам MS-DOS, UNIX, языку Си, аппаратному обеспечению и сопроцессорам.

Восемь раз в год тиражом 2200 экземпляров выходит клубный журнал под названием "User" (пользователь). На его страницах печатаются только статьи членов AUGE. Авторы получают гонорар за публикации - до 100 марок за страницу, остальная работа в клубе не оплачивается. Члены AUGE получают журнал бесплатно, вне Ассоциации он не распространяется.

Некоторые члены AUGE организуют на своих компьютерах "электронные конференции" - Bulletin Board Systems (BBS). В ближайшем будущем AUGE создаст свои собственные узлы в информационных сетях Zerberus-net и FIDOnet. Это направление считается одним из самых важных в деятельности AUGE. Уже сейчас многие члены Ассоциации имеют свои идентификаторы в различных коммерческих сетях.

Большая библиотека некоммерческого программного обеспечения также привлекает пользователей компьютеров различных типов. Для того чтобы бесплатно получить такие программы, надо обращаться к президентам специальных групп по интересам (согласно международной практике некоммерческое программное обеспечение запрещено продавать; за процесс копирования, однако, взимается умеренная плата, обычно 2-4 марки за одну дискету).

Члены AUGE приобретают компьютеры и периферийное оборудование со скидкой - это также один из привлекательных моментов членства в клубе.

Руководит клубом исполнительный комитет, избираемый каждые два года. В ближайшее время планируется объединение всех европейских компьютерных клубов в одну Ассоциацию, что, несомненно, ускорит обмен информацией. AUGE открыта для советских пользователей и надеется на установление деловых контактов. Международный компьютерный клуб, который был представлен на конференции двумя советскими представителями, намерен способствовать этому процессу.

\ в мире "Компьютера" \

© Владимир Федоров

Болгарский клон Apple II на советском рынке

Для большинства непрофессиональных советских пользователей понятие персонального компьютера ассоциируется с машинами фирмы IBM: XT и AT прочно вошли в нашу жизнь. Многие считают, что только эти ПК являются "настоящими" компьютерами. В таком подходе ярко проявляются традиции однозначного, унифицированного подхода, длительное время формировавшегося в Советском Союзе. Сначала были машины, похожие на LSI и PDP фирмы DEC (семейство "Электроника"), затем, с созданием процессора KP5801K80 (аналога Intel 8080), появилась плеяда компьютеров с операционной системой CP/M-80 и, наконец, на советский рынок хлынул поток IBM-совместимых машин.

Однако, несмотря на солидное положение IBM, в мире успешно сосуществуют десятки других персональных компьютеров. Компьютеры Apple II являются одними из наиболее популярных, а фирма Apple Computer, Inc. прочно стоит в первой десятке крупнейших американских компьютерных корпораций.

В 1982 г. в небольшом городке Правец, что в 70 км от Софии, началось строительство крупнейшего в Восточной Европе комбината микропроцессорной техники. Судя по многомиллионным валютным затратам и размаху предприятия, Т. Живков активно поддерживал идею превращения своей родной деревни в болгарскую "кремниевую долину", причем руководили проектом весьма грамотные люди. В США была полностью закуплена фирма по производству компьютеров и все оборудование перевезено в Правец. Ходил анекдот, что в последнем контейнере были обнаружены веники и коробки с мусором - в Соединенных Штатах остались лишь стены. Были куплены акции ряда западноевропейских и сингапурских компьютерных фирм - дивиденды давали возможность закупать комплектующие детали за рубежом. Первым компьютером КМПП "Правец" стал аналог популярнейшей американской машины Apple II+ - "Правец-82".

Разработчики компьютера "Правец-82" - сотрудники института технической кибернетики и робототехники (ИТКР БАН) Иван Марангозов и Петр Петров не ошиблись в выборе платформы. До сих пор 61% компьютеров в системе образования США - компьютеры Apple II. В 1990 г. было торжественно объявлено о выпуске пятимиллионного компьютера Apple II, а сколько еще компьютеров выпустили фирмы Franklin Computer, Laser Computer и др.? Американцы справедливо полагают, что эпоха персональных компьютеров началась с Apple II. Сейчас мир Apple II - это более 20000 программных пакетов, сотни периферийных плат, десятки независимых компаний, поддерживающих машины такого типа. Это - проводящаяся два раза в год международная выставка AppleFest, десятки журналов, ассоциаций пользователей, системы электронной почты и объявлений (Bulletin Board Systems - BBS).

За четыре года КМПП "Правец" буквально завалил Болгарию удобными и дешевыми компьютерами. Поставки в СССР блокировались из-за спланированного "разделения труда" в рамках СЭВ - как экспортер персональных компьютеров Болгария не значилась. Положение изменилось в 1988 г. с созданием в Ташкенте совместного советско-болгарского предприятия "Вариант". Его генеральный директор Григорий Бровман сумел за короткий срок организовать сборку компьютеров "Правец-8А" из болгарских комплектующих частей. СП "Вариант" нацелено на рынок компьютеров для школы. Его основная продукция - компьютерные классы - установлена практически во всех школах ПТУ и техникумах г. Ташкента. За два года Узбекистан в принципе решил проблему с "железом" - в республике более 50000 компьютеров "Правец-8А". Стандартная конфигурация учебного класса "Вариант" состоит из 11 компьютеров, соединенных кольцевой локальной сетью. Рабочее место учителя имеет 4 дисковод и принтер. СП дает гарантию на один год, обеспечивает техническое обслуживание в течение семи лет и берет на себя сопровождение машин, включающее поддержку программным обеспечением и обучение пользователей. Разработаны программные пакеты по основным школьным курсам: математика, физика, геометрия и т.д.

С созданием своих филиалов и технических центров по всей территории СССР СП "Вариант" расширяет и географию своих поставок: Москва, Ленинград, Вильнюс, Нижний Новгород, Тольятти, Львов и т.д. Один класс (11 машин) стоит около 60000 руб. - приблизительно столько же, сколько и один компьютер IBM PC/AT. К сожалению, помимо школ установлено пока лишь 1000 машин, так как большинство поставок происходит под жестким контролем Госкомитета по народному образованию.

Чем же привлекателен "Правец-8А" для пользователей? Прежде всего хорошим соотношением производительность/стоимость. Действительно, не каждая организация может себе позволить купить для секретаря с окладом 300 руб. в месяц рабочее место за 50000 руб. Автоматизация управления, бухгалтерская деятельность, учет на средних и мелких предприятиях - вот весьма перспективные области применения этой надежной и простой машины.

"Парис" (болгарская адаптация интегрированного пакета AppleWorks американской фирмы Claris) - текстовый процессор/база данных/электронная таблица - для ПК "Правец-8А" быстро завоевал популярность в СССР. AppleWorks до сих пор является бестселлером на рынке программного обеспечения. Только за три месяца AppleWorks "продал" полмиллиона компьютеров Apple II, а его создатель Боб Лисснер (Bob Lisner) стал таким же знаменитым, как и создатели Apple II - Стив Джобс (Steve Jobs) и Стив Возняк (Steve Wozniak).

В настоящее время на советском рынке находится от 70 до 90 тыс. компьютеров Apple II различных моделей. СП "Вариант" создало советский парк Apple-совместимых машин, но было ограничено в возможностях создать рынок этих компьютеров, так как основные поставки были сделаны в сферу народного образования, а школы, как известно, не очень богатые организации.

"Не кооперированные" пользователи (т.е. находящиеся вне системы народного образования) смогли вздохнуть свободнее лишь в 1989 г., когда НПО "Тандем" на базе московского физико-технического института начало продажу отдельных компьютеров. Это объединение специализируется на поставке автоматизированных рабочих мест "под ключ" - с программным обеспечением, "настроенными" базами данных, электронными таблицами и т.д. Виктор Колесник, генеральный директор "Тандема", сделал еще один шаг вперед к "цивилизованному" рынку: в Москве открыт фирменный магазин НПО "Тандем", где можно приобрести не только компьютеры "Правец-8А" (6000 руб.), но и дополнительные периферийные платы, программное обеспечение и запасные принадлежности. Магазин обслуживает как организации, так и отдельных лиц.

Очевидно, что для успешного распространения компьютера необходимы два условия: развитое программное

обеспечение и самостоятельные Ассоциации пользователей. Секция A2-Link Международного компьютерного клуба (МКК) уже объединила независимых пользователей и разработчиков компьютеров Apple II/"Правец" не только в СССР, но и за рубежом. A2-Link имеет исключительное право на распространение в СССР американского журнала "The Road Apple", обменивается публикациями с ведущими "яблочными" журналами - InCider, A2-Central. МКК абонируется в международной электронной информационной сети GENIE (General Electric Network Information Exchange). На электронном "круглом столе" этой сети (Apple Round Table) собираются сообщения пользователей Apple из многих стран мира. Дружественные контакты установлены с европейской Ассоциацией пользователей Apple (Apple Users Group Europe - AUGЕ), ведущими западными фирмами - Roger Wagner Publishing, Beagle Brothers, Claris, Stone Edge Technologies и т.д.

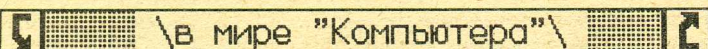
В июне 1990 г. в московском Центре международной торговли прошел первый Международный компьютерный форум (МКФ), организованный МКК, на котором состоялась презентация фирмы Apple Computer. Директор Управления новых разработок Кирк Ловнер (Kirk Loevner) был приятно удивлен интересом советских пользователей к компьютерам фирмы. В отличие от многочисленных компьютерных выставок-ярмарок, ориентированных на сбыт вычислительной техники, МКФ стал местом непринужденного общения, обмена опытом, лекций и семинаров крупнейших специалистов. Для участия в Форуме в Москву прибыли президенты таких фирм, как WordPerfect Corp., Adobe Systems, Borland International.

Одной из тем МКФ, вызвавшей повышенный интерес, стало положение о соблюдении авторского права (copyright) в СССР. Рынок программных продуктов в стране отсутствует, и попытка придерживаться законов об охране авторского права приводит к необходимости искать валюту для закупки программных средств. Первым шагом для выхода из создавшейся ситуации может стать легальная продажа лицензированных и адаптированных западных программ на советском рынке за рубли.

Для реализации этого проекта МКК уже предпринял необходимые усилия. В октябре 1990 г. начнется распространение пакета Micol Advanced Basic для компьютеров "Правец-8" и Apple II. Micol Advanced Basic - структурированный компилируемый язык, разработан в 1988 г. канадской фирмой Micol Systems Inc. Язык создан специально для компьютера Apple, поддерживает все его графические режимы (вплоть до 560 x 192, 16 цветов) и популярен в американских школах. Документация на русском языке содержит 300 страниц и поставляется вместе с дискетой и регистрационной карточкой. Стоимость комплекта, определенная лицензионным соглашением между Micol Systems и МКК - 180 руб. (99 дол. в США). Зарегистрированные (т.е. "легальные") пользователи будут иметь техническую поддержку по двум телефонным номерам, скидку при приобретении следующих версий Micol Basic, бесплатную подписку на бюллетень секции A2-Link, а также станут получать сообщения обо всех изменениях в программе и информации о новых версиях и программных продуктах.

Интересно, предотвратит ли умеренная стоимость нелицензированное копирование продукта? Будущее покажет, не предпочтут ли наши пользователи "бесплатные" трудности!





© Иван Оловенцев, Игорь Щетинин

Какие Спектрымы * ходят в Союзе

В 1980 г. произошло знаменательное событие – был выпущен первый домашний компьютер (home computer) фирмы **SINCLAIR RESEARCH**, довольно быстро снижавший популярность.

Это был дешевый и очень удобный в обращении аппарат, работающий с обычным черно-белым телевизором. В 1982 г. фирма выпускает новую модель – ZX Spectrum, имеющую все черты современного компьютера и позволяющую формировать на экране телевизора цветное графическое изображение, подключать принтер, сетевой адаптер и прочие периферийные устройства. При том, что эта модель оставалась достаточно дешевой, при желании ее можно было использовать не только для игр, но и для более серьезных целей. Огромное количество программ, написанных для Спектрума (в настоящее время исчисляющееся несколькими тысячами), и предельно «дружелюбный» Бейсик делают его незаменимым в «домашнем хозяйстве».

Начиная с середины 80-х гг. волны компьютеризации стали докатываться и до нашей страны, еще только начинающей оправляться от кубика Рубика. Радиолюбители каким-то образом умудрялись доставать «антисоветские» детали и собирать Спектрымы по появившимся «пиратским» схемам. Об этих схемах и хочется повести разговор.

Начнем со схемы, получившей название «Московская». Она насчитывает 68 корпусов и по духу наиболее близка к оригиналу. Однако сейчас эта схема практически забыта, так как на печатной плате требовалось вносить много исправлений и не у всех хватало на это терпения. Некоторых отпугивала достаточно дефицитная (и потому дорогая) по тем временам комплектация. Последующие версии шли по пути упрощения этой схемы и некоторые «зашли» даже очень далеко. Но какой ценой?

Известная схема «Балтика» с целью уменьшения количества микросхем была построена с помощью запоминающих ИС на плавких перемычках, а также дополнена портом 580BB55A, что делало удобным подключение к ней дополнительных устройств. Использование нестандартного для Спектрума кварцевого генератора на 16 МГц позволяет упростить подключение контроллера дисководов. Однако это же мешает нормально пользоваться программами, которые используют рамку (BORDER) для каких-либо рисунков. Любой рисунок на рамке оказывается в самых неожиданных местах и в самом непредсказуемом виде. Во всех остальных моделях этот вопрос решается чуть лучше, т.е. рисунок на рамке виден, однако он не совмещается по местонахождению с рисунком на экране (PAPER). Связано это с тем, что синхронизация изображений происходит благодаря приходу импульса на вход маскируемого прерывания процессора Z80. Этот импульс должен приходить

в момент появления первой строки экрана (PAPER) и длиться около 9,5 мкс, в то время как практически во всех «самопальных» моделях прерывание происходит по кадровому синхроимпульсу, а в некоторых, вдобавок ко всему, еще и не предусмотрено гашение.

Очень интересное решение этой проблемы предложено в версиях «Зеленоград» и «Красногорск», являющихся счастливым исключением. Здесь в видеоконтроллере использовано ПЗУ 573PФ2, в котором программно записаны временные диаграммы для прерываний, гашения, синхроимпульсов и управления экраном. Несогласные с автором всегда могут переделать содержимое ПЗУ на свой вкус. Заложенные в «Красногорск» и «Зеленоград» принципы похожи, поэтому относительно подробно рассмотрим только «Зеленоград». Эта схема интересна тем, что в ней уже четко прослеживается стремление привести правильность работы в жертву простоте. В частности, в схеме нет и намека на порт FF, а остатки декодировки портов клавиатуры, джойстика и рамки больше напоминают рудименты и атавизмы, нежели авторский замысел. Правда, на плате, по счастью, оказываются свободными один вентиль 2И-НЕ и два вентиля 2ИЛИ, и это все-таки позволяет без чрезмерных усилий подключить порт 580BB55A. Из очень положительных моментов этой схемы отметим, что в младшие 16 Кбайт ОЗУ можно из Бейсика записать любой другой язык и, отключив ПЗУ, работать в его среде. «Красногорск» выбрал в себя все положительные черты «Зеленограда» и те немногие достоинства, которые чудом оказались в «Ленинграде». Вообще стоит отметить проработанность разводок печатных плат «Красногорска» и «Зеленограда» – все связи выведены на разъемы, число доработок уже в первых вариантах разводок минимально, а в последних они вообще отсутствуют. Схожесть недостатков этих версий и «Ленинграда» наводит на мысль о родстве душ их разработчиков. Сейчас трудно ответить на вопрос, что помешало автору «Красногорска» сравняться с питерскими коллегами – самолюбие или чувство юмора, но разработчики «Ленинграда» оказались куда настойчивее и последовательнее.

Схема «Ленинград» является на сегодняшний день самой популярной и самой неправильно работающей. Добиваясь невиданной до этого дешевизны и компактности, ее авторы, видимо, ставили перед собой задачу испортить все, что было хорошего во всех предыдущих схемах. Этот компьютер годится только для тех, кто собирается играть целыми днями, не помышляя ни о каком ином применении машины, так как имеющиеся в модели грубые ошибки делают проблематичным любое иное ее использование. В схеме отсутствует какая-либо адресация любых портов ввода-вывода, и если вы решитесь расширить ее, то на вашу голову обрушится масса неприятностей. При попытке что-либо записать в дополнительно подключенное устройство рамка начнет усиленно моргать, а динамик верещать «дурным голосом». Попытка что-либо считывать может окончиться еще плачевнее, ибо при этом на шину данных будет поступать информация не только от вашего устройства, но и от его соперника – микросхем 555КП11. И совсем не факт, что жертв и разрушений при этом не окажется. Еще более серьезным недостатком, раздражающим уже во время игры, является некорректно построенный видеоконтроллер. Как известно, в строке должно уместиться 32 знакоместа; однако счетчик столбцов может считать только до 31 столбца. Чтобы обойти этот недостаток, в видеоконтроллер введена RC-цепочка, вносящая задержку в момент обнуления счетчика. Из-за этого крайний столбец отличается от остальных.

В последнее время заслуженной популярностью начинает пользоваться схема, совмещающая на одной плате как Спектрум, так и контроллер дисководов и получившая звучное название «Пентагон». Достоинство такого подхода очевидно – избавившись от дисководов, вам не нужно будет сначала ломать голову над выбором схемы контроллера и искать соответствующую плату, а потом тянуться к плате паяльником сквозь лианы соединительных проводов. Кроме того, в схеме уже заложено подключение порта 580BB55A. Правда, она также не лишена целого ряда не-

* Так мы будем называть клонированную модель ZX Spectrum и ее советские аналоги.

достатков: в компьютерной части отсутствует буферирование между шиной адресов и клавиатурой, при попытке записи в младшие 16 Кбайт адресного пространства на шину данных выдается также содержимое соответствующей ячейки ПЗУ, выходные видеосигналы подаются ТТЛ-уровнем без возможности их регулировки и учета яркости; в контроллере дисководов наиболее ярким "глюком" является попытка использования микросхемы 561LN1 (6 КМОП-инверторов с третьим состоянием) в качестве "буфера-защиты". Мало того, что пришлось последовательно включать 2 инвертора и вносить большую задержку, эта микросхема не в состоянии "прокачать" перегруженную и поднятую резисторами к +5 В шину данных в отсутствие двунаправленного буфера 580BA86 (такая ситуация возможна при использовании ПЗУ типа 27256). Выйти из положения можно, усилив выходные буферы за счет 2 неиспользованных и подключив их параллельно штатным. Также непонятно, почему разработчики не привели разъем дисководного интерфейса в соответствие со стандартом.

Заканчивая обзор схем, хотелось бы пофантазировать, каким должен быть "идеальный Спектрум": он должен иметь системный разъем и правильную дешифрацию портов ввода-вывода, разъем контроллера дисководов должен быть совместим со стандартным. Он обязательно должен иметь порт FF, ибо в противном случае защита программ от "пиратских" машин организуется до смешного тривиально. Желательно также дополнить машину контроллером принтера, причем не того, который используется в комплекте Спектрума, а фирменного интерфейса, имеющего программное обеспечение (например, "Tasman"). Если же вы не хотите ждать, пока увидит свет такой Спектрум, то начинающим рекомендуем "Красногорск", людям с опытом советуем обратить внимание на "Пентагон". Пусть не вполне идеально, но они все-таки работают.

Авторы благодарят Константина Мусатова за обсуждение статьи и дружеские советы.



В мире "Компьютера"

© Юрий Храмов

Средства создания экспертных систем

Экспертные системы (ЭС), появившиеся в конце шестидесятых годов в качестве исследовательских проектов (MUSIN, DENDRAL), к настоящему времени стали одной из серьезных коммерческих отраслей информационной индустрии. Рынок программного обеспечения в этой области стабильно и быстро расширяется. По различным оценкам западных специалистов, в 1990 г. его объем составит от 0,9 до 1,4 млрд. дол. Уже при создании первых ЭС было отмечено, что механизм логического вывода и язык представления знаний (ЯПЗ) могут быть использованы в различных проблемных областях. Так появилась идея разработки "пустых" проблемно-независимых ЭС - экспертных систем с незаполненной базой знаний. Такие системы получили название **ОБОЛОЧЕК ЭКСПЕРТНЫХ СИСТЕМ** (expert system shells). Для получения прикладной экспертной системы

пользователь должен создать свою собственную базу знаний, используя предоставляемый оболочкой ЯПЗ. Манипуляция знаниями, генерацию объяснений, а также сервис разработки и отладки базы знаний обеспечивают встроенные средства оболочки. Хотя ряд ЭС разработан непосредственно на специализированных языках программирования (Лисп, Пролог, OPS5) или даже на "обычных" языках (Си, Паскаль), в настоящее время основную часть продаж на рынке ЭС обеспечивают именно оболочки. Оболочки ЭС создаются и для "больших" машин, и для машин серии VAX, и для многих других (в том числе дорогих специализированных станций искусственного интеллекта), но наиболее массовый рынок - это персональные компьютеры, совместимые с IBM PC. Многие оболочки, первоначально созданные на других типах машин (в том числе на Лисп-машинах), сейчас доступны на ПК. Именно поэтому в обзоре рассмотрены наиболее распространенные оболочки для ПК типа IBM PC.

1. **CxPERT, Software Plus** (\$795), США. Главной отличительной чертой системы является то, что при создании базы знаний допускается свободное использование, наряду с конструкциями Cxpert, фрагментов, написанных на Си. Полученный текст обрабатывается препроцессором и преобразуется в исходный текст на Си. Механизм вывода, генератор объяснений и другие части системы представляют собой библиотеку Си-программ, которая используется при построении исполнимого модуля ЭС. Такой подход обеспечивает очень высокую гибкость системы, возможность писать очень сложные программы, встраивать экспертные системы в большие программы на Си и/или вызывать свои программы в "нужном" месте консультации.

2. **Exsys 3.0, Exsys** (\$395), США. Написанная на Си относительно несложная система продукционного типа с обратным методом вывода. Также поддерживается механизм типа "классной доски" для выделенных модулей знаний (экспертных подсистем), которые могут передавать данные, однако только через файл на диске. Допускается применение факторов уверенности. Фирма предоставляет также **Exsys Professional** (\$795), имеющую дополнительно командный язык, управляющий вводом данных, применением правил, организацией циклов и выводом данных. Имеются средства доступа к файлам dBaseIII+ и Lotus 1-2-3. Тесты показывают, что обе эти системы весьма эффективны и справляются с базами знаний больших объемов. Система достаточно популярна на американском рынке, но существенно уступает VP Expert и Level5.

3. **GoldWorks II, Gold Hill Computers** (\$8900), США. Очень мощная ЭС, сочетающая в себе различные формы представления знаний.

Используемые в системе термины "фреймы" (frames) и "примеры" (instances) достаточно точно соответствуют терминам "класс" и "объект" в объектно-ориентированных языках программирования. Слоты фреймов могут содержать и данные, и ссылки на другие фреймы или примеры, и некоторые процедуры. Так называемые фасеты слотов (facets) определяют методы доступа и установления значений, способы наследования и т.п.

Правила системы в посылке содержат утверждения относительно значений слотов примеров или фреймов, а в заключение - передачу сообщений фреймам или примерам для установления значений слотов или каких-либо иных действий. Допустимо использование пропозициональных переменных. Правила организованы в группы, для каждой из которых могут быть указаны условия активизации и стратегия логического вывода (прямой, обратный или двунаправленный вывод). Имеются средства для имитации параллельной работы нескольких групп правил, поддержки немонотонной логики, задания факторов уверенности.

GoldWorks позволяет использовать окна, меню, графику (в том числе динамическую) для организации пользовательского интерфейса, обеспечивает гибкий доступ к файлам dBase и Lotus 1-2-3. Разработчик может воспользоваться "интеллектуальным" редактором правил, графически





ми средствами просмотра фреймов и других структур базы знаний (в том числе в процессе отладки). Система имеет оригинальное средство импорта знаний - ASCII parser, которое позволяет транслировать тексты и таблицы в структуры языка GoldWorks, предварительно описав схему синтаксического разбора. GoldWorks написана на Лиспе (GC Lisp) и допускает встраивание Лисп-фрагментов в базы знаний. Последняя версия имеет также очень удобные интерфейсы с Microsoft и Lattice C.

В целом надо отметить, что достаточно трудно представить себе задачу, непосильную для средств GoldWorks II. К недостаткам системы можно отнести ее перегруженность понятиями, относительно слабые (по сравнению с другими свойствами) средства объяснений и высокие требования к техническому обеспечению (PC/AT, 4 MB RAM, предпочтительно 386-й процессор). Система достаточно популярна в США, используется многими серьезными компаниями как для решения задач управления производством, так и в банковских системах. На персональных компьютерах GoldWorks опережает своих основных конкурентов (Nexpert и KEE), зато KEE наиболее популярен на рабочих станциях и ЭВМ VAX.

4. Guru, MDBS (\$6500), США. Не только и не столько оболочка, сколько интегрированный пакет для одновременной работы с базами данных и знаний. ЭС этой системы - продукционного типа, с наличием прямого и обратного вывода и развитым механизмом обработки неточных знаний и управления ходом вывода. Кроме ЭС пакет включает мощную реляционную СУБД, электронную ведомость, средства деловой графики, текстовый редактор, средства планирования экранных и печатных форм и естественно-языковой интерфейс баз данных. Выдвинутый фирмой принцип синергизма позволяет использовать в любом месте практически любые средства системы. Например, применить консультацию экспертной системы для пересчета клеток электронной ведомости. Встроенный процедурный язык системы обладает достаточной мощностью, чтобы реализовать на базе Guru и фреймовый подход, в том числе даже наследование, запоминание и использование опыта, а также организовать удобный интерфейс конечного пользователя в процессе консультаций и объяснений, хотя все эти средства "штатно" не предусматриваются. На сегодняшний день такая степень интеграции различных компонент уникальна, система не имеет в этой области конкурентов и позволяет встроить в традиционные АРМы фрагменты искусственного интеллекта. Платой за огромные возможности является сложность системы, усугубляемая низким качеством документации.

5. KDS3, KDS Corp. (\$1495), США. Оболочка, целиком написанная на Ассемблере. Основное достоинство системы - генерация базы знаний с помощью процедуры, напоминающей игру "животные". Хотя обычно в подобной игре строится бинарное дерево решений, разработчики утверждают, что KDS строит гораздо более сложный объект - четырехмерную матрицу переходов, использует многозначную логику и т.п. Мне, однако, обнаружить эти свойства не удалось. Построенное дерево может быть отредактировано пользователем для введения дополнительных условий, связей, группировки вопросов и т.п. Организация системы позволяет динамически расширять базу знаний, и алгоритм ее построения оказывается очень простым для пользователя. Скорость и простота разработки содержательных баз знаний с помощью этой системы впечатляют. Система имеет жесткий, хотя и весьма удобный интерфейс текстового режима, сопровождается средством "film maker", для проектирования графических экранов. Знания могут быть развиты на модули, что позволяет строить очень большие и высокоскоростные ЭС. Имеются прекрасные интерфейсы с внешними программами.

6. KEE386, IntelliCorp. (\$9900), США. Среда построения ЭС, первоначально созданная для работы на станциях искусственного интеллекта. В 1988 г. была выпущена версия

для Compaq Deskpro386, в настоящее время имеется версия для IBM PS/2 модель 80. Все они требуют очень больших размеров ОП (более 6 Мбайт). Основной набор возможностей достаточно близок к GoldWorks II, но имеет несколько большие возможности работы с фреймами; в частности, допускается множественное наследование. Зато отсутствует "интеллектуальный" редактор правил, не поддерживается работа с факторами уверенности. Система обладает средствами тестирования и отладки знаний, прекрасным пользовательским интерфейсом, включающим графику с высокой степенью разрешения. Система написана на Lucid Common Lisp и требует установки на компьютер ОС UNIX System V.

7. KnowledgePro, Knowledge Garden (\$495), США. Написанная на Турбо Паскале оболочка среднего класса, использующая процедурный язык для описания процесса консультации. Все атрибуты, используемые при консультации, должны быть описаны с помощью так называемых тем (topics). Темы объединяются в дерево описания консультации с применением наследования для доступа к значениям и процедурам. Для вывода значений внутри темы используется обратный метод вывода, а для комментирования выделенных пользователем значений - средства гипертекста, что вполне удачно заменяет отсутствующую в KnowledgePro подсистему объяснений. Обеспечивается доступ к файлам dBase и Lotus 1-2-3. Фирма распространяет также дополнительные утилиты Knowledge Maker (\$99), превращающие файлы 1-2-3 в правила типа IF-THEN, и Knowledge Graphic Toolkit, позволяющий применять в создаваемой прикладной системе графические возможности KnowledgePro - относительно новая система, в настоящее время активно улучшающая свои позиции на рынке и сейчас не слишком сильно отстающая от признанных лидеров рынка - VP Expert, Level5, EXSYS.

8. Level5, Information Builders (\$685), США. Продукционная система, ориентированная на обратный вывод, однако имеющая и средства прямого вывода. Имеется язык представления продукционных правил, средства обработки неточных знаний. Предусмотрена компиляция базы знаний перед консультацией. Фирма рекламирует "полную интегрированность с базами данных" (на IBM PC - dBaseIII+) и высокую скорость работы. Система очень популярна, реализована на различных компьютерах и является одним из основных претендентов на роль стандарта небольшой оболочки для IBM PC. Совсем недавно фирма выпустила новую версию (Level5 Plus), обладающую возможностями гибридных систем, и повела активное наступление на позиции Nexpert Object в этой области. Созданы также системы приобретения знаний QUEST и KAT, порождающие правила в формате Level5. Эти системы основаны на тех же идеях, что и KDS, и при построении деревьев решений способны оценивать их состоятельность.

9. Nexpert Object, Neuron Data (\$3000), США. Мощное средство разработки ЭС, конкурирующее с такими системами, как GoldWorks II и KEE, и имеющее подобную организацию базы знаний (см. GoldWorks). Nexpert на IBM PC работает в среде MS Windows, и разработчик получает доступ к средствам построения интерфейсов этой системы. Фреймы допускают деление слотов на наследуемые и ненаследуемые, использование "демонов", динамическое установление связей между фреймами в процессе консультации. Система имеет интерфейсы с языками программирования, хорошие возможности работы с базами данных, поддерживает SQL-запросы таких СУБД, как dBase, Oracle, Ingres и др. Система имеет средства объяснений (why, how), специализированный меню-ориентированный редактор для создания баз знаний.

В отличие от своих конкурентов система написана не на Лиспе, а на Си, что позволило снизить требования к ЭВМ: для полной версии системы достаточно IBM PC/AT с 4 Мбайтами ОП, упрощенные версии могут быть использованы на ЭВМ Macintosh с 512 Кб ОП. В сочетании с относительно невысокой стоимостью это делает систему весьма перспективной. Фирма разработала и специальную систему для извлечения знаний - Nextra. Она ведет диалог с экс-

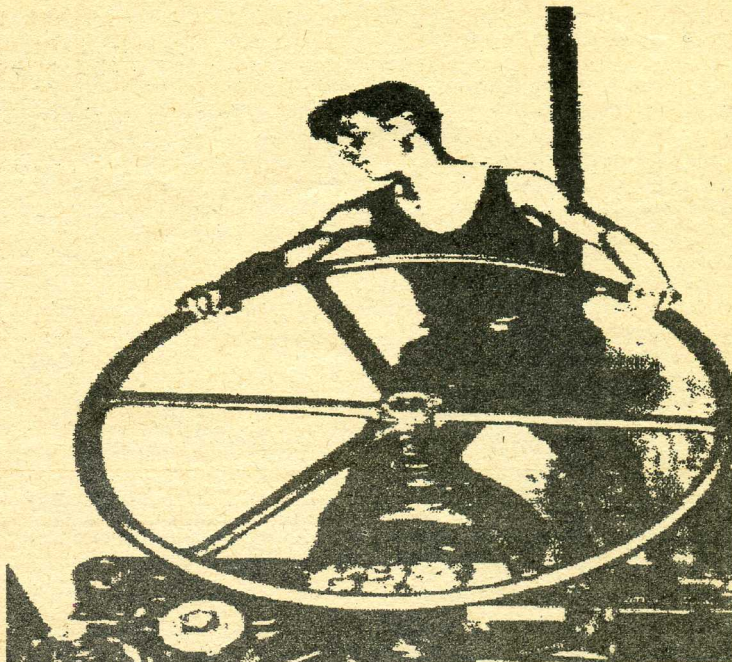
пертом для извлечения знаний, некоторым образом интегрирует их и затем представляет в графической форме для редактирования. Теоретической базой диалога служит метод репертуарных решеток. Кроме того, имеются индуктивные средства генерации правил, классов и объектов из фактов, мнений и примеров (cases), представленных экспертами. В настоящее время Nextra реализована для Mac Plus, II и SE и стоит около \$4000.

10. Personal Consultant Plus, Texas Instruments (\$2950), США. Система имеет средства прямого и обратного вывода, фреймовое представление знаний. Однако в данной системе фреймы - не аналог объектов, как в GoldWorks, а скорее средство структурирования множества правил. Возможны внешние вызовы, доступ к файлам dBase, в том числе выбор записей по условию и исполнение программ dBaseIII+, метаправила, механизм наследования, расширение возможностей системы за счет построенных пользователем Лисп-функций. Для реализации всех возможностей требуется 2 Мбайта ОП (причем управление ими обеспечивает сама система), но большой набор функций может быть реализован и на PC XT, а так называемая gultime-версия способна работать даже на машине без винчестера. Наряду с PC+ фирма поставляет и упрощенную версию системы - PC Easy (\$495), не поддерживающую фреймы. Кроме того, имеется программа Procedural Consultant (\$495), позволяющая строить дерево решений в режиме диалога. Полученное дерево может быть потом превращено в набор правил для PC Easy или PC Plus.

С учетом этих расширений Personal Consultant - очень широко распространенная система, но если два года назад она была безусловным лидером рынка, то сейчас ее положение колеблется.

11. Rule Master, Radian (\$595), США. Гибкое средство создания ЭС с использованием механизма индукции правил. При создании БЗ требуется задать список гипотез (возможных решений ЭС), таблицу примеров, состоящую из наборов значений признаков и соответствующей гипотезы, а также, возможно, процедуры получения значений признаков. На основании этой информации Rule Master генерирует программу на встроенном языке программирования Radial, которая способна проводить консультацию. Для генерации правил используется метод индуктивного построения правил ID3. Полученный текст ЭС можно редактировать и расширять, при этом Radial обладает довольно широкими возможностями. Кроме того, Rule Master способен по таким же таблицам генерировать исходный текст программы на Си, что позволяет встраивать полученную программу в более сложные системы или вызывать из ЭС программы на Си. Последняя версия Rule Master способна генерировать сложные запросы к базам данных.

12. VP Expert, Paperback Software (\$249), США. Большинство экспертов сходятся на том, что по соотношению эффективности и цены VP не знает себе равных. Этот компактный пакет имеет много средств, выходящих за пределы обычно представленных в подобного рода изделиях системы продукции. VP Expert может работать с переменными, что позволяет преодолеть ограничения логики исчисления высказываний; разработаны очень простые и разумные средства работы с базами данных в стандарте dBase, VP Info и с электронными таблицами в формате Lotus 1-2-3. В систему встроены механизмы запоминания ситуации, фрагментирования БЗ, средства организации многоцелевой консультации. VP Expert содержит также ограниченный внутренний язык, позволяющий создавать несложные процедуры, управлять диалогом с пользователем. Имеется библиотека математических функций. Система представляет собой легкий для освоения и разработки ЭС. VP имеет средства генерации правил по записям базы данных, названные в документации индуктивным обучением. Новая версия системы (2.0) включает средства поддержки окон, гипертекст для организации более содержательного диалога с пользователем, рудиментарную объектно-ориентированную систему и средства для встраивания в Си-программы.



Основной недостаток VP - невозможность работы с объемными СУБД: переполнение внутреннего стека вывода происходит очень быстро, и базы приходится разбивать на сектора. VP Expert - лидер на рынке оболочек экспертных систем, но основное направление ее применения - создание демонстрационных систем, а не промышленное использование.

13. 1stClass HT, 1stClass Expert System (\$2495), США. Система, генерирующая правила по примерам, заданным в форме таблиц. Имеются средства коррекции построенных деревьев решений, графическое представление правил, средства структурирования проблемы. Для индуктивной генерации правил система, как и остальные подобные западные средства, использует алгоритм ID3 Куинлена (режим оптимизации дерева). Возможна также простейшая стратегия построения дерева, ручное заполнение или корректировка. Можно отказаться от построения дерева, тогда программа при оценке нового примера будет искать наиболее похожий из существующих. Кроме обычных средств для индуктивных систем, имеются дополнительные возможности: создание и использование гипертекстов, простой интерфейс с программами на процедурных языках, доступ к данным в формате dBaseIII+, Lotus 1-2-3, а также средства управления режимом консультаций. По построенным деревьям решений 1st Class способна создать исходные тексты программ на Си или Паскале. Версия системы 1stClass Fusion (\$1495) не поддерживает работу с гипертекстами, но позволяет добавлять новые факторы как логическую или арифметическую комбинацию существующих. Сейчас 1stClass по объему продаж превосходит не только любую индуктивную ЭС, но и большинство "стандартных" оболочек ЭС.

Приведенные системы составляют только небольшую часть доступных на рынке оболочек. Из остальных отметим производственные оболочки Crystal (Intelligent Environment, \$995) PC Expert и PC Expert Pro (Software Artistry, \$295 и \$495 соответственно), английские оболочки Flex и Xi Plus, индуктивные системы Logic Gem, Logic Tree, PC/Beagle и TIMM, "набор" средств фирмы Intelligence Ware (IXL, I/C), позволяющий генерировать правила на основе информации (таблиц) баз данных, мощное средство генерации экспертных систем KES II фирмы Software A&E, очень перспективную систему OPS200 фирмы Intellipro (\$295), ART IM PC фирмы Inference (\$8000).

Сравнительная оценка и выбор наиболее подходящего продукта - сложная проблема для любого класса программного обеспечения. Даже для программ с очень близкими





функциями и возможностями (например, компиляторов одного языка) существуют разные методики, оценки ведутся по различным критериям и выводы их не совпадают. При сравнении продуктов с такими разнообразными функциями, как оболочки ЭС, трудно надеяться даже на выбор какой-либо объективной системы тестов, поэтому единственным методом остаются экспертные оценки. Чтобы упростить оценку и сделать ее результаты более объективными и понятными пользователям, надо классифицировать свойства оболочек ЭС. К сожалению, в этой области также не выработана общепринятая точка зрения. Предлагаемый ниже набор свойств не претендует на исключительность и полноту, однако, по моему мнению, позволяет отражать три главные проблемы, с которыми сталкивается разработчик, выбирая, а затем используя оболочку:

- потенциальные возможности системы;
- легкость разработки баз знаний;
- эффективность и удобство работы конечного пользователя с построенными прикладными системами.

В первую группу входят:

A1. Выразительная сила языка представления знаний. Это свойство дает возможность описать с помощью ЯПЗ системы сложные логические связи в БЗ, используя штатные (и потому компактные, относительно эффективные и понятные) средства системы. На одном полюсе здесь находятся системы, допускающие описание в БЗ только правил вида

ЕСЛИ $A=a$ и $B=b$ и ..., ТО $C=c$,

где $a, b, \dots, c$ - константы (так называемая пропозициональная логика). На другом - системы, полностью реализующие логические возможности исчисления предикатов или даже логик более высоких порядков.

A2. Средства управления логическим выводом - часто их называют метазнаниями или метаправилами. Чем гибче стратегия использования знаний в системе, чем больше у нее "знаний о знаниях", тем больше результатов система может получить из одного и того же объема знаний, тем шире становится круг вопросов, на которые она может ответить, тем выше информативность и обоснованность ответов, а также эффективность работы системы.

A3. Структуризация знаний - характеристика того, насколько имеющиеся в ЯПЗ структуры позволяют отразить структуру знаний в предметной области. Если гибкие структуры ЯПЗ позволяют достаточно адекватно отразить структуризацию знаний эксперта, заполнение БЗ будет осуществляться значительно легче, БЗ будет более понятной и читаемой. Кроме того, наличие структур в ЯПЗ может помочь эксперту структурировать свои знания. Как правило, структурные возможности дают оболочки, использующие фреймы и семантические сети, объектно-ориентированный подход к языку представления знаний.

A4. Процедурные расширения. Очень часто выбранная вами оболочка содержит много средств, разработчику не нужных, и не содержит какого-нибудь небольшого, но принципиально важного для данного приложения фрагмента. Наиболее естественный путь в таком случае - немного расширить возможности системы самому. То, насколько это возможно и просто, и определяет открытость системы. Достичь этого можно различными путями: обеспечением возможности вызывать внешние программы, встраивать в БЗ фрагменты, написанные на языке разработки самой оболочки, изменить свойства оболочки при доступности исходных текстов. Мощный встроенный язык программирования также способен обеспечить необходимые процедурные расширения.

A5. Связь с базами данных. В настоящее время говорить о разработке серьезных ЭС без связи с базами данных не приходится. Практически в любой области достаточно много знаний накоплено в виде данных, и в серьезной системе наряду с интеллектуальными функциями воз-

никает необходимость обрабатывать данные традиционным образом - готовить отчеты и выборки из базы данных, осуществлять поиск, просмотр и другие функции, присущие СУБД.

Вторая группа представлена следующими свойствами:

B1. Средства приобретения знаний. Средства автоматизированного расширения БЗ представляют исключительную важность для характеристики оболочки, так как основная доля трудоемкости при создании ЭС падает как раз на работу по получению знаний. В коммерческих системах подобные средства представлены возможностями накопления опыта и построением правил по индукции на основе примеров, а также средствами генерации дерева решений в диалоге с пользователем.

B2. Сервис создания и отладки БЗ. Важность этих средств объясняется теми же причинами, что и в предыдущем пункте. При оценке мы исключили из рассмотрения то, что входит в понятие "приобретение знаний". Здесь оценивается мощность и качество специализированных редакторов БЗ, средства синтаксического и логического анализа БЗ, качество трассировки вывода, средства просмотра текущего состояния системы и т.п.

B3. Легкость изучения. Важность этого свойства признается для всех классов программных продуктов, но для оболочек ЭС, где не существует стандартов, концепции довольно сложны и необычны для пользователя, легкость изучения просто критична. Естественно ожидать, что чем проще система, тем легче ее освоить. Однако наличие хороших обучающих программ, подробных примеров, а главное, удачно написанная документация могут существенно упростить освоение системы. Напротив, неудачная документация (как, например, в Guru) может привести к необходимости больших затрат на освоение системы.

В третью группу включены:

C1. Качество пользовательского интерфейса. Здесь следует оценивать две вещи - насколько удобен и гибок предоставляемый системой стандартный интерфейс и насколько сильны средства создания собственного пользовательского интерфейса: можно ли использовать окна, меню, графику и другие подобные средства.

C2. Читаемость БЗ и качество объяснений. Общая направленность этих двух характеристик - оценить простоту работы системы. Разумное использование результатов консультации возможно только в том случае, если система может хорошо объяснить свои выводы; легко читаемая БЗ позволяет понять, почему система действует именно так, и в случае необходимости скорректировать базу.

C3. Эффективность и мощность. Здесь оцениваются предельно допустимые размеры БЗ и время реакции системы. Отметим, что для ЭС время реакции не имеет такого критического значения, как, скажем, для СУБД, и следует опасаться лишь катастрофического замедления при относительно небольших размерах БЗ.

Оговоримся, что приводимые свойства "неортогональны": многие из них коррелируют друг с другом, а одна и та же функция оболочки может оказать влияние на оценки по целому ряду свойств.

В заключение приведу таблицу оценок свойств тех оболочек, с которыми я имел возможность познакомиться на практике.

	a1	a2	a3	a4	a5	b1	b2	b3	c1	c2	c3
Guru v1.1	хор	отл	уд	хор	отл	слаб	уд	слаб	хор	уд	хор
VP Ex. 1.0	хор	хор	слаб	слаб	хор	уд	уд	хор	хор	уд	уд
1stClass	уд	уд	слаб	уд	слаб	отл	хор	хор	уд	слаб	?
PC+ v2.0	хор	отл	хор	хор	хор	слаб	слаб	хор	хор	хор	уд
Level5	уд	хор	слаб	хор	хор	хор	уд	хор	хор	хор	отл
EXSYS	уд	уд	слаб	уд	уд	нет	уд	уд	уд	хор	отл
KDS v2	уд	уд	уд	хор	?	хор	хор	отл	хор	слаб	отл?
Rule Mast.	уд	слаб	слаб	хор	нет	хор	уд	уд	слаб	слаб	хор

ГЛОССАРИЙ

БАЗА ЗНАНИЙ (БЗ) экспертной системы – формализованное описание знаний о проблемной области. Формализм, используемый для ввода человеческих знаний в ЭВМ, называется **ЯЗЫКОМ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ (ЯПЗ)**. Иногда из БЗ выделяют **БАЗУ ФАКТОВ (БФ)**, содержащую данные об объектах проблемной области, и собственно БЗ, содержащую правила использования фактов для получения новых фактов в качестве ответов на запросы пользователя. С некоторой точки зрения БЗ можно рассматривать как программу, написанную на ЯПЗ.

МЕТАЗНАНИЯ – информация о том, как организованы знания в ЭС – знания о знаниях. Используются для управления ходом логического вывода. Одним из наиболее распространенных видов метазнаний служат **МЕТАПРАВИЛА** – продукционные правила, действия которых изменяют стратегию механизма логического вывода.

МЕТОД КЛАССНОЙ ДОСКИ – способ организации знаний, при котором база знаний разбивается на независимые модули. Каждый модуль “наблюдает” за своей областью классной доски и включается в работу только тогда, когда там появляется новая информация. В процессе работы модули оставляют полученные ими новые знания в других областях.

МЕХАНИЗМ ЛОГИЧЕСКОГО ВЫВОДА – часть экспертной системы, обеспечивающая получение ответов на вопросы пользователя на основе знаний, содержащихся в БЗ. Данный механизм должен обеспечивать подобие рассуждений человека и может генерировать в процессе вывода новые знания (факты). Для систем, основанных на продукционном представлении правил, используется **ПРЯМОЙ, ОБРАТНЫЙ** или **СМЕШАННЫЙ** механизм логического вывода.

НЕЧЕТКИЙ ВЫВОД – механизм логического вывода, используемый при наличии нечетких знаний. В продукционных системах он чаще всего основан на использовании так называемых **ФАКТОРОВ УВЕРЕННОСТИ**. Каждому правилу и каждому значению используемого в БЗ объекта (факту) может быть приписано некоторое число, определяющее степень уверенности в истинности правила или значения. Когда в процессе логического вывода система создает новые факты, и факторы уверенности рассчитываются на основе факторов уверенности посылок и самого правила. Обычно системы допускают использование различных методов пересчета факторов уверенности.

ОБОЛОЧКИ ЭКСПЕРТНЫХ СИСТЕМ – ЭС с незаполненной базой знаний, т.е. содержащие механизм логического вывода, ЯПЗ, средства генерации объяснений и организации интерфейса. Используются как средства разработки ЭС.

ОБРАТНЫЙ ВЫВОД – один из механизмов логического вывода, используемый в продукционных системах. Системе задается конечная цель, которую необходимо вывести из исходных фактов. Для этого: 1) определяются продукции, с помощью которых можно достигнуть искомую цель; 2) проверяется истинность посылок в БЗ; 3) посылки, истинность которых не подтверждается и не опровергается БЗ, объявляются новыми целями, после чего про-

изводится возврат к пункту 1. Такой возврат происходит до тех пор, пока все необходимые для вывода цели не окажутся выполненными на БЗ либо пока не исчерпаются правила, подходящие для дальнейшего вывода.

ПРОДУКЦИИ, продукционные правила – наиболее известный формализм представления знаний в ЭС. Продукция, или продукционное правило, представляет собой пару “условие” → “действие”. Смысл ее состоит в том, что при выполнении условия система должна выполнить действие, указанное в правой части продукции. Условие должно использовать информацию о текущем состоянии базы знаний, а действия могут изменять это состояние и/или сообщать результаты пользователю.

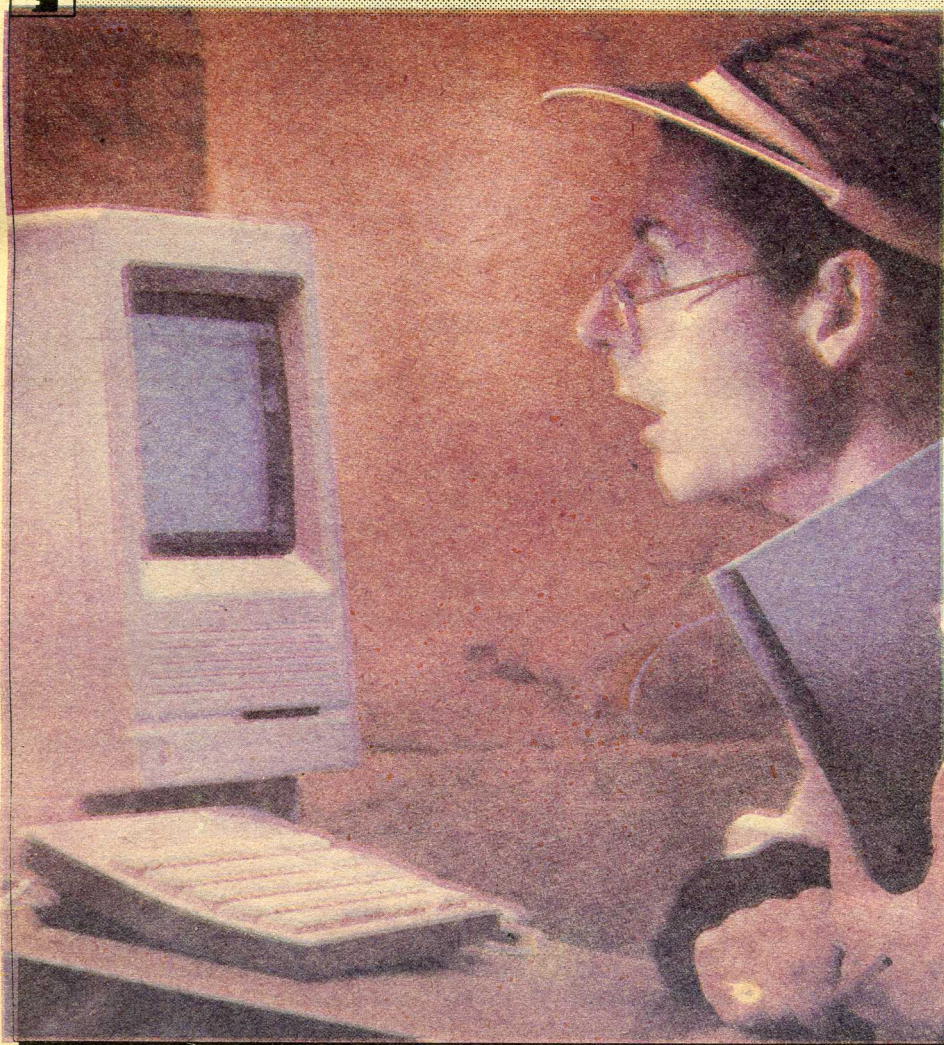
ПРЯМОЙ ВЫВОД – один из механизмов логического вывода, используемый в продукционных системах. Сводится к следующей процедуре: 1) определяются все правила, условия которых выполняются при текущем состоянии базы знаний (БЗ); 2) с помощью стратегии “разрешения конфликтов” выбирается некоторое из них (например, первое по списку); 3) выполняются действия, связанные с выбранным правилом, что приводит к изменению базы знаний; 4) если имеются правила, условия которых выполняются в измененной БФ, переходим к пункту 2.

СЕМАНТИЧЕСКИЕ СЕТИ – формализм представления знаний в ЭС. Согласно этому методу знания представляются в виде некоторых объектов, связанных различными отношениями типа “является частью”, “аналогичен” и т.п. или более специфичными, например “работает в”, “состоит в браке с”. Выписывание всех существенных для рассматриваемых объектов отношений и представляет собой семантическую сеть. Объекты служат узлами сети, отношения представляются дугами.

ФРЕЙМЫ – один из распространенных формализмов представления знаний в ЭС. Фрейм можно представить себе как структуру, состоящую из набора ячеек – слотов. Каждый слот состоит из имени и ассоциируемых с ним значений. Значения могут представлять собой данные, процедуры, ссылки на другие фреймы или быть пустыми. Такое построение оказывается очень удобным, в частности, для моделирования аналогий, описания областей с родо-видовыми связями понятий и т.п.

ЭКСПЕРТНЫЕ СИСТЕМЫ (ЭС) – программы, предназначенные моделировать работу человека-эксперта при решении некоторой сложной задачи. Главными частями ЭС являются **БАЗА ЗНАНИЙ (БЗ)**, содержащая формализованные знания о той проблеме, для решения которой она создана, так называемый **МЕХАНИЗМ ЛОГИЧЕСКОГО ВЫВОДА**, который должен обеспечивать подобие рассуждений человека на базе этих знаний, а также **ПОДСИСТЕМА ОБЪЯСНЕНИЙ, ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ**. Отличительными особенностями ЭС является возможность объяснить полученное решение, средства работы с неточными и неполными знаниями (**НЕЧЕТКИЙ ВЫВОД**).

ЯЗЫК ПРЕДСТАВЛЕНИЯ ЗНАНИЙ (ЯПЗ) – средства формального представления знаний о проблемной области, используемые при составлении **БАЗЫ ЗНАНИЙ**. Основными формализмами, используемыми современными коммерческими ЭС, служат **ПРОДУКЦИИ, ФРЕЙМЫ, СЕМАНТИЧЕСКИЕ СЕТИ**.



- 📁 IBM-совместимость
... на СМ1810?
- 📁 Методы защиты
от вирусов
- 📁 "Прогулки"
по MS Word
- 📁 OPEN ACCESS II
- 📁 Новинки
FIDO
- 📁 Переписка
с читателями

📁 \компьютер на работе\ 📁
© Константин Коробов

IBM-совместимость ... на СМ1810?

С Игорем Скориковым, главным инженером СКТБ ПО "Орловский завод УВМ им. К.Н. Руднева", во время демонстрации новых (61 и 62) моделей СМ1810, серийно выпускаемых заводом с начала 1990 г., беседовал Константин Коробов.

К.К. Игорь Петрович, по внешней компоновке и наблюдаемым на экране знакомым программам (НС и др.) ваши компьютеры, особенно настольный вариант 62-й модели, действительно похожи на РС XT. Однако после публикации в 1 выпуске нашего сборника статьи "Что такое IBM-совместимость?" даже ранее непосвященные читатели вправе задаться вопросом, о какой IBM-совместимости (объявленной заводом) идет речь в данном случае? Другими словами, смогут ли пользователи вашей машины без проблем работать со всеми популярными программами, привычными на IBM PC XT/AT?

И.С. Сначала нужно сказать несколько слов о самой машине. Речь идет о "персоналках" с тактовой частотой 5 МГц и быстродействием не менее 1 млн. оп/с, сопроцессором INTEL 8087, с памятью ОЗУ 1 Мбайт, 20-мегабайтовым

жестким диском ST-225 сингапурского производства и дисководом для пятидюймовых дискет 360 Кбайт. Эти ПК укомплектованы цветным монитором "Электроника" MC-6106 (по моему мнению, лучшим из производимых в стране) с разрешением 640x200 точек в цветном и 640x400 точек в монохромном режиме, а также EPSON-совместимым принтером с широкой кареткой советского производства. Клавиатура, разработанная фирмой IBM, и лишь чуть-чуть "втиснутая" в советские ГОСТы (расположение клавиш в верхнем регистре, как на пишущей машинке), поставляется вместе с драйвером, ориентированным на КОИ-7. Как видите, внешние параметры стандартны для IBM-совместимой машины.

Теперь об уровне совместимости (об операционной системе, BIOS и аппаратных особенностях). Разработчики машин - сотрудники ИНЭУМа - стремились получить максимальную совместимость с прототипом IBM PC XT и одновременно расширить их функциональные возможности. Цель была достигнута за счет использования интерфейса И-41 (MULTIBUS-1), обеспечивающего достаточно простое подключение дополнительных устройств к комплексам. Операционная система МДОС1810 аналогична MS-DOS 3.30. Проводилась проверка работы компьютеров с ОС MS-DOS 3.30 и MS-DOS 4.0, показавшая достаточно надежную работу с различными программами. Оригинальный BIOS близок к варианту, применяемому в прототипе. Однозначно можно утверждать, что цель, поставленная ИНЭУМом, достигнута. Если все-таки пользователи встретятся с проблемами, то мы всегда готовы помочь решить их.

К.К. Советский рынок постепенно наполняется IBM-совместимыми ПК, широко продаваемыми теперь и за руб.



ли. Какие преимущества ваших машин оставляют Вам надежды на предпочтения заказчиков?

И.С. Во-первых, наши цены, вернее ценовая эффективность комплектов, предпочтительнее для заказчиков. Самая скромная в отношении возможностей расширения конфигурации модель 62 стоит 22 тыс. руб., в случае поставки с тремя дополнительными дисплеями для многопользовательской работы с разделением ресурсов (для не слишком ресурсоемких задач) ее цена возрастает до 30 тыс. руб. Кстати, любая наша поставка соответствует специфическим потребностям заказчиков. При разделении жесткого диска на области возможна работа с разных терминалов под управлением любой из операционных систем: МДОС1810 (MS-DOS V3.3), БОС-1810 (RMX-86), DEMOC (UNIX V1.6). Для обеспечения многозадачного многопользовательского режима в ОС МДОС1810 используется комплекс программ типа "MULTILINK". С середины 1991 г. мы планируем поддержку сетей (типа Token-Ring, Ethernet), причем будет обеспечена возможность интеграции в сеть "разношерстных" ПК: SM1810, Нейрон, импортных машин.

Таким образом, наши "козыри" это - комплектная поставка систем "под ключ", обеспеченность техническими средствами для диагностики и ремонта, а также обучение пользователей в фирменном учебном центре.

К.К. Можно догадаться, что и в этом случае речь идет об адаптации зарубежных сетевых программных продуктов, к особенностям аппаратуры, так же как и в случае с упомянутыми Вами операционными системами. В этой связи возникает деликатный вопрос о лицензионной чистоте ваших разработок и об авторском праве на программы. Чувствуете ли вы себя достаточно защищенными, и если да, то на чем основана ваша позиция?

И.С. При разработке наших компьютеров во главу угла была поставлена проблема обеспечения совместимости с прототипом на уровне операционной системы и прикладных программ. Эта задача нами решена. Таким образом, пользователь, работающий с фирменными продуктами, затруднений не встречает. К настоящему моменту в СССР создано достаточное количество организаций, имеющих права на распространение фирменных программных продуктов. Наше объединение также очень заинтересовано в получении прав на продажу фирменных продуктов, но, как только речь заходит о рублях, разговоры на этом заканчиваются. Мы же свою продукцию продаем за рубли. Именно так и решается этот действительно деликатный вопрос.

К.К. Не связаны ли некоторые особенности ваших новых машин с традициями завода, многие годы выпускавшего управляющие машины для автоматизации производства?

И.С. Да, и это тот случай, когда "доставшееся наследие", вопреки привычным нашему поколению понятиям, имеет хороший смысл. Поддержка непрерывных технологических процессов предъявляет ужесточенные требования к надежности машины. Стендовые испытания 10 серийных образцов SM1810 (модели 61, 62) продемонстрировали наработку на отказ не менее 5000 ч. Для этих моделей помимо возможностей широкого гуманитарного применения, предусмотрены интерфейсные места для обработки информации с выносных модулей, решающих задачи автоматизации различных промышленных предприятий - от элеваторов до прокатных станов. Учитывая возможные условия производственной эксплуатации машин упомянутые показатели надежности обеспечиваются при "тропическом" температурном режиме. Предусмотрены также специальные вентиляционные фильтры для блока центрального процессора.

К.К. Скажите, не собираетесь ли вы еще "круче" развернуться к гуманитарным потребностям экономики? В буклетах, распространенных на презентации новых SM1810, есть упоминание о планах производства недорогих 8-битовых компьютеров и графических станций, по-видимому, пригодных и для издательских нужд?

И.С. Вы абсолютно правы. С 1991 г. мы начинаем поставлять компьютерные классы на 16 мест на базе 8-битовых SM1810 модели 70 (операционная среда CP/M), укомплектованные дисковыми для 5-дюймовых дискет, винчестерами и стримерами, работающими в сети, "посаженной" на SM1810 модели 70 или 62 в качестве рабочего места преподавателя информатики и программирования. Один компьютерный класс будет стоить около 40 тыс. руб. Во второй половине 1991 г. мы планируем начать выпуск машин с развитыми графическими возможностями, поддерживающими работу с программами САПР и верстки текстов. Для издательских нужд спроектирована система с цветным монитором и разрешением 1024x736 точек (типа АРТИСТ-1), позволяющая осуществлять подготовку текстов и графики в MS-Word 5.0 и выводить непосредственно диапозитивы для печати. Ориентировочная стоимость комплекта 100 - 300 тыс. руб., в зависимости от варианта исполнения и комплектности.

К.К. Мой последний вопрос о ваших ближайших производственных и коммерческих планах. Какие усовершенствования претерпят ваши "персоналки" и как будет налажен их сбыт?

И.С. Что касается усовершенствований, то это - одноплатный контроллер жесткого диска, центральный процессор с тактовой частотой 8 МГц, стримеры с контроллерами. Уже есть заказы на поставку. В случае, если портфель заказов окажется слишком "пухлым" для производственных возможностей орловского завода, мы готовы наладить кооперацию и увеличить выпуск до десятков тысяч ПК в год. Грядущий рынок заставит нас заниматься маркетингом и думать об интересах пользователей. Для этой цели мы создаем сеть представительских центров, где можно посмотреть и потрогать руками наши машины, получить консультации и справочные буклеты, оформить заявки. В ближайшем будущем эти центры будут выполнять пусконаладочные работы, гарантийное обслуживание и послегарантийный ремонт.

К.К. Благодарю Вас за беседу.

* Заявки и вопросы заинтересованных предприятий и организаций могут быть направлены также в редакцию журнала "Компьютер".

Комплект документации для пользователей IBM PC

Многие из вас помнят книгу В.Э.Фигурнова "IBM PC для пользователя", выпущенную издательством "Финансы и статистика" в 1 квартале 1990 г. В настоящее время автор значительно ее расширил, дополнил дискетой, и теперь этот комплект документации и программ под названием "Работа пользователя с IBM PC" распространяется по предприятиям и организациям. В состав комплекта входит документация (640 страниц) и дискета (360 Кбайт) с программным обеспечением и полным комплектом шрифтов для редактора ChiWriter, в том числе лазерных.

В документации содержатся практически все сведения, необходимые для пользователей PC. В частности, полностью описаны команды MS-DOS, DR DOS, Norton Commander версии 3.0, редакторы текста Лексикон и ChiWriter, все программы комплекса Norton Utilities, разновидности компьютерных вирусов и программ для защиты от них, средства конфигурирования системы, программы и методы для восстановления удаленных файлов и многое другое. Вместе с тем комплект может служить хорошим учебным пособием для пользователей, в том числе начинающих.

Для приобретения комплекта следует направить гарантийное письмо в адрес редакции (101000, Москва, ул. Чернышевского, 7, редакция журнала "Компьютер"). Цена 1 экз. комплекта - 790 руб. (для вузов - 590 руб.). При приобретении нескольких экземпляров на все экземпляры, кроме первого, скидка 40%.

Методы защиты от вирусов

Какие средства применяются для защиты от вирусов? Для этого можно использовать:

- общие средства защиты информации, которые полезны также и как страховка от физической порчи магнитных дисков, неправильно работающих программ или ошибочных действий пользователей;
- профилактические меры, позволяющие уменьшить вероятность заражения вирусом;
- специализированные программы для защиты от вирусов.

Антивирусная профилактика рассматривалась в первом выпуске сборника "Компьютер" (с. 49), поэтому здесь мы эти проблемы обсуждать не будем.

Общие средства защиты информации полезны не только для защиты от вируса. Есть две основные разновидности этих средств: копирование информации - создание копий файлов и системных областей дисков; разграничение доступа предотвращает несанкционированное использование информации, в частности защиту от изменений программ и данных вирусами, неправильно работающими программами и ошибочными действиями пользователей.

Несмотря на то, что общие средства защиты информации очень важны для защиты от вирусов, все же только их недостаточно. Необходимо и применение специализированного программного обеспечения. Эти программы можно разделить на несколько видов: детекторы, вакцины (иммунизаторы), доктора (фаги), ревисоры (программы контроля изменений в файлах и системных областях дисков), доктора-ревизоры и фильтры (резидентные программы для защиты от вирусов). Поскольку в предыдущих выпусках сборника эти виды программ уже обсуждались, то сейчас мы приведем лишь краткие определения этих понятий.

Программы-детекторы позволяют обнаружить файлы, зараженные каким-либо одним известным вирусом или одним из нескольких известных вирусов.

Программы-вакцины, или иммунизаторы, модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, от которого производится вакцинация, считает эти программы или диски уже зараженными.

Программы-доктора, или фаги, лечат зараженные программы или диски, "выкусывая" из зараженных программ тело вируса, т.е. восстанавливая программу в то состояние, в котором она находилась до заражения вирусом.

Программы-ревизоры сначала запоминают сведения о состоянии программ и системных областей дисков, а затем сравнивают их состояние с исходным. О выявленных несоответствиях сообщается пользователю.

Доктора-ревизоры - это гибриды ревизоров и докторов, т.е. программы, которые не только обнаруживают изменения в файлах и системных областях дисков, но и могут в случае изменений автоматически вернуть их в исходное состояние.

Программы-фильтры, или резидентные программы для защиты от вирусов, постоянно располагаются в оперативной памяти компьютера. Они перехватывают те обращения к операционной системе, которые используются вирусами для размножения и нанесения вреда и сообщают о них пользователю. Пользователь может разрешить или запретить выполнение соответствующей операции.

Сравнение средств защиты от вирусов

Многие пользователи компьютерной техники считают, что для защиты от вирусов следует собрать как можно больше программ, обнаруживающих и уничтожающих вирусы (т.е. программ-детекторов и докторов), а остальными мерами защиты можно пренебречь: когда появится вирус, тогда из этих программ наверняка можно будет выбрать подходящее "лекарство". Между тем для минимизации ущерба от вируса надо пользоваться правилом, которое давно исповедуют медики: болезнь легче предупредить, чем вылечить. В связи с этим дадим краткое описание различных средств защиты от вирусов с учетом их вклада в безопасность программ и данных. Предлагаемые сравнительные оценки являются, разумеется, личным мнением автора.

Первое место по вкладу в безопасность компьютеров и данных, безусловно, занимает копирование данных. Без периодического копирования данных не должен работать ни один серьезный пользователь. При заражении вирусом еще можно восстановить хотя бы часть данных, но что вы будете делать, если на компьютере сломается жесткий диск? Кроме того, при любом повреждении программ и данных, копии которых имеются в архиве, наиболее целесообразно не пытаться проводить их лечение различными "докторами", а восстановить их из архива.

На второе место следует поставить разграничение доступа к данным. Если большинство используемых пакетов программ находится на логическом диске с защитой от записи, то при заражении вирусом эти пакеты не будут повреждены и вам потребуется гораздо меньше усилий для устранения последствий заражения. Кроме того, все архивные дискеты надо снабжать наклейкой защиты от записи. Такой подход к обеспечению безопасности можно сравнить с правилом, давно применяемым в кораблестроении - даже если один отсек корабля дал течь, остальные должны держаться на плаву.

На третьем месте находятся программы-ревизоры (программы раннего обнаружения заражения вирусом), позволяющие определять целостность программ и данных. Такая проверка позволяет выявить наличие вируса на самом раннем этапе, когда он еще не успел все испортить. Проверку целостности желательно проводить каждый раз в начале работы с компьютером, т.е. включить ее в файл AUTOEXEC.BAT. Программа FluShot Plus позволяет проверять целостность программ также при их запуске на выполнение. Кроме того, с помощью программы-ревизора можно определить, какие файлы были заражены или испорчены вирусом. Очень удобно, когда программа-ревизор может не только лечить файлы при наиболее распространенных видах их заражения, но и восстанавливать загрузочный сектор и сектор с таблицей разбиения жесткого диска при порче вирусами, распространяющимися через загрузочный сектор.

На четвертое место я бы поставил программы-фильтры (резидентные программы для защиты от вируса) типа Anti4us и FluShot Plus. Эти программы позволяют обнаружить многие (хотя и не все) вирусы на самой ранней стадии, когда они не успели еще что-либо заразить или испортить.

На пятом месте находятся программы-детекторы. Их очень часто применяют совершенно неправильно, вставляя в файл AUTOEXEC.BAT команды запуска таких программ для проверки всего жесткого диска на заражение программами вирусами. Такую проверку гораздо лучше выполнит программа-ревизор, а программы-детекторы надо использовать для проверки вновь полученного программного обеспечения на наличие вирусов.

Шестое место (а вовсе не первое) занимают программы-доктора (фаги). Их лучше применять только тогда, когда копии зараженной программы нет в архиве, и ее трудно получить иным образом. Кроме того, если уж вы применяете программу-фаг, то потом обязательно проверьте восстановленный файл программой-ревизором (разумеется, если информация об этом файле была предварительно сохранена) - отнюдь не всегда программы-доктора "лечат" правильно.

И наконец, на последнее место надо поставить программы-вакцины. В мире всего существует несколько сотен вирусов, поэтому вероятность защитить файл именно от того вируса, которым будет заражен компьютер, ничтожно мала. И кроме того, более эффективно поместить программу на защищенную от записи дискету или на защищенный от записи логический диск.

Стратегия защиты от вирусов

Ни одно из описанных выше средств не дает, к сожалению, полной защиты от вирусов. Поэтому наилучшей стратегией защиты от вирусов является многоуровневая, "эшелонированная" оборона. Опишем структуру этой обороны против вирусов.

Средствам разведки в "обороне" от вирусов соответствуют программы-детекторы, позволяющие проверять вновь полученное программное обеспечение на наличие вирусов.

На переднем крае обороны находятся программы-фильтры (резидентные программы для защиты от вируса). Эти программы могут первыми сообщить о работе вируса и предотвратить заражение программ и дисков.

Второй эшелон обороны составляют программы-ревизоры, программы-доктора и доктора-ревизоры. Ревизоры обнаруживают нападение вируса даже тогда, когда он сумел "просочиться" через передний край обороны. Программы-доктора применяются для восстановления зараженных программ, если ее копий нет в архиве, но они не всегда лечат верно. Доктора-ревизоры обнаруживают нападение вируса и лечат зараженные программы, причем делают это правильно.

Самый глубокий эшелон обороны - это средства разграничения доступа. Они не позволяют вирусам и неверно работающим программам (даже если они проникли в компьютер) испортить важные данные.

И наконец, в "стратегическом резерве" обороны находятся архивные копии информации и "эталонные" дискеты с программными продуктами. С их помощью можно восстановить информацию при ее повреждении.

Будем надеяться, что это неформальное описание позволит лучше понять методику применения антивирусных средств.

компьютер на работе

© Григорий Сенин

"Прогулки" по MS Word

Описание системы **Microsoft Word** в первом прошлогоднем номере "Компьютера" было сродни красочному путеводителю, в лучшем случае - рассказу о чужом путешествии... На этот раз мы сами "прогуляемся по окрестностям" MS Word, чтобы убедиться в преимуществах и недостатках системы, не принимая на веру рекламные заявления.

С областью обработки текстов так или иначе сталкивались очень многие. Большинство, по-видимому, усвоило ее понятия и получило соответствующие навыки, пользуясь распространенным текстовым процессором Лексикон, разработанным Е. Веселовым. Отталкиваясь от этой практики, мы и попробуем перейти в область более "высокоразвитой технологии", которую представляет собой MSW. И первое, на что следует обратить внимание, - это формат файлов.

Формат файлов

Документ, который вы создаете и обрабатываете, текстовый процессор или система подготовки текстов хранят в файлах того или иного формата. Программа Лексикон, как и некоторые другие редакторы, оперирует с файлами наиболее простого вида, которые можно назвать чистым текстом. В таком файле документ отражен буквально, буква в букву. Всего лишь несколько особых знаков, кроме обычных, могут содержаться в текстовом файле; среди них - символ, называемый "переводом строки". Благодаря этому символу при выводе на экран или принтер файл выпадает разбитым на отдельные строки. Другое наименование чисто текстовых файлов - ASCII-файлы (по названию стандартной таблицы символов).

Хранение текста в простом текстовом формате имеет известные преимущества главным образом потому, что этот формат является определенным стандартом для операционной системы. С помощью одной команды DOS текстовый файл легко вывести на экран, склеить с другим таким же файлом, напрямую послать на принтер и получить распечатку. Такие файлы служат естественной формой внутри- и межкомпьютерной передачи информации. Наконец, такой формат поддается обработке самыми разными программами-редакторами.

Все хорошо до тех пор, пока мы не вспомним о внешнем виде текста, пока речь не идет о его оформлении с целью получения качественной печати. Задумываясь об этом, мы поймем, почему с точки зрения развитой текстовой обработки чистый текст не может быть стандартом. В лучшем случае это стандарт "для бедных". Пока мы ограничиваемся лишь содержанием документа, нам достаточно считать его "просто текстом". Но как только речь заходит об интервале печати, отступах, о курсивном, полужирном и прочем выделении символов - тогда собственно текста нам становится мало, потому что о некоторых его элементах (знаках, словах, абзацах) требуется дополнительная информация. Неудивительно, что структура файла MSW отлична от ASCII-файлов - ведь помимо самого текста приходится хранить и информацию о его оформлении.

Журнал "Бухгалтерский учет" объявляет Всесоюзный конкурс на лучший пакет программ по автоматизации бухгалтерского учета для:

- государственных предприятий и организаций;
- совместных предприятий;
- кооперативов;
- малых предприятий.

Более подробную информацию об условиях конкурса можно узнать по адресу:

103006 Москва, ул. Садово-Триумфальная, 4/10, журнал "Бухгалтерский учет".

Справки по телефонам:

299-86-70 Александр Сладков
299-20-12 Дмитрий Белов

Поскольку файл MSW устроен иначе, сложнее, чем чистый текст, его уже не удастся в нормальном виде вывести на экран командой ДОС. (Нельзя сказать, что на экране будет полная бессмыслица, но... нечто довольно бессвязное.) Кроме того, увеличится и объем файла (коэффициент расширения при "богатом" оформлении может превышать 1.5).

Было бы не совсем правильным считать, что оформление текста - прерогатива высокоразвитых текстовых процессоров и вовсе недоступно программам класса Лексикон. Любой пользователь заявит, что это не так. Многие элементы оформления - выравнивание, красная строка, поля, интервал - обеспечиваются и в Лексиконе. Но возникают вопросы, какими средствами это достигается и насколько полно обеспечиваются возможности оформления?

Программа-редактор, которая ограничивается простым текстовым форматом, как это делает Лексикон, для форматирования волей-неволей прибегает к средствам редактирования, к средствам "не того уровня". В результате некоторые символы приходится нагружать дополнительной "форматирующей" функцией. Примеры можно найти в том же Лексиконе: пробел здесь используется для создания "левого поля" и "красной строки", "минус" употребляется для переноса слов, а для увеличения интервала печати вставляются лишние строки. Очевиден "приближенный" характер такого форматирования, когда отступы измеряются в пробелах, интервалы - в строках. Но не менее важен и второй момент: возможности подобного совмещения функций весьма ограничены и реализовать с его помощью все разнообразие оформления текстов нельзя.

Предшествующие размышления тесно связаны с практической задачей. Если Лексикон и MSW по-разному "смотрят" на текст, то как преодолеть различие взглядов? На примере этой задачи хорошо видны несходство "средней" и высокоразвитой текстовой обработки и та граница, по которой проходят принципиальные различия.

Как привести файл в формат MSW?

Если вы изначально работаете в MSW, то проблем, разумеется, не возникнет - программа сама проследит за тем, чтобы по окончании работы файл был сохранен в нужном виде (если вы сами не захотите изменить формат). А как быть, если вам достался файл, "сработанный" в Лексиконе? Казалось бы, ничего страшного: Word умеет читать такие файлы. Однако если попробовать работать в MS Word прямо с текстовым файлом, то нас постигнет разочарование.

не так. Ровный правый край текста вдруг нарушится, каждая строка в абзаце начнет вести себя независимо от других. Впечатление будет такое, что не работает автоподверстывание (рис. 2). В чем дело? Все объясняется, когда вы узнаете, что каждая строка воспринята программой как отдельный абзац. Это не случайно, такое поведение программы имеет свое объяснение.

Начать надо с того, что разбивка на абзацы и разбивка на строки (мы говорим, естественно, о прозе) имеют мало общего; только машинистка не видит между ними особой разницы. Абзацы в тексте выделяет автор, руководствуясь смыслом своего произведения. Разбивка же на строки автору в сущности безразлична; это элемент чистой формы, в котором много случайного и который зависит главным образом от ширины листа бумаги и других внешних факторов.

Наиболее употребительной информацией в базах данных являются числа (целые и вещественные) и строки. Объектная база данных комплекса РеСпект поддерживает следующие дополнительные типы данных, которые могут служить характеристиками объектов (полями записей):

а) Текст. В виде атрибута объекта может храниться текстовая информация большого объема (около 16 Кбайт):

в текстовом поле отдельные слова могут быть отмечены как ключевые - по ним обеспечивается быстрый поиск объектов. При создании и модификации текстовых записей специальными маркерами в текстах могут выделяться слова и выражения, которые становятся ключевыми и позволяют впоследствии производить быстрый поиск этих текстовых записей в режиме карточки.

б) Текстовый файл. Произвольный текстовый файл может быть объявлен значением поля записи; обеспечивается просмотр файла из базы данных. Это позволяет использовать готовую информацию, ранее введенную в традиционном текстовом редакторе.

Pg1 L17 Co54 (до 18) ? ZH TEST.LEX Microsoft Word

Рис. 2. "Несовпадение взглядов": при внесении в файл правки красивая картина нарушается

Word "играет" по вполне естественным правилам: "покажите мне, где в тексте абзацы, т.е. как вы изложили содержание, а о форме, о том, как разбить его на строки, позабочусь я". Иначе говоря, программа исходит из того, что если текст разбит на фрагменты, то это - абзацы. Символами разбиения Word считает как раз "переводы строки", аккуратно расставленные в тексте Лексиконом. Так-то вот.

"Позвольте, - скажет читатель. - Почему программа облегчает себе жизнь? Абзацы снабжаются и чисто формальными признаками, например красной строкой. Разве не может программа сама их обнаруживать?". Как ни странно, ответ будет отрицательным.

Дело в том, что внешние очертания текста (ровные края, отступы, красные строки), легко отмечаемые глазом человека, программе, которая текста "не видит", воспринять гораздо труднее. Если в текстовом файле имеется общий отступ от левого края (левое поле), то для программы отдельные строки мало чем отличаются: это цепочки символов с несколькими пробелами в начале. Текст, представленный на рис. 1, может показаться довольно регулярным. Однако он типичен в том отношении, что содержит абзацы разных очертаний (меняется отступ от края, "нависает" красная строка). С абсолютной точностью определить границы абзацев со столь разными внешними признаками программе затруднительно.

Итак, можно констатировать основное неудобство при загрузке в MSW текстового файла: "рассыпанные" на отдельные строки абзацы необходимо восстановить в их первоначальном виде.

Каждая строка текстового файла, воспринятая MSW как содержимое отдельного абзаца, в дальнейшем может неограниченно расти, как цепочка символов. Это не мешает ее

Наиболее употребительной информацией в базах данных являются числа (целые и вещественные) и строки. Объектная база данных комплекса РеСпект поддерживает следующие дополнительные типы данных, которые могут служить характеристиками объектов (полями записей):

а) Текст. В виде атрибута объекта может храниться текстовая информация большого объема (до 18 Кбайт): в текстовом поле отдельные слова могут быть отмечены как ключевые - по ним обеспечивается быстрый поиск объектов. При создании и модификации текстовых записей специальными маркерами в текстах могут выделяться слова и выражения, которые становятся ключевыми и позволяют впоследствии производить быстрый поиск этих текстовых записей в режиме карточки.

б) Текстовый файл. Произвольный текстовый файл может быть объявлен значением поля записи; обеспечивается просмотр файла из базы данных. Это позволяет использовать готовую информацию, ранее введенную в традиционном текстовом редакторе.

Pg1 L11 Co1 () ? ZH TEST.LEX Microsoft Word

Рис. 1. Файл Лексикона загружен в MS WORD: Все в порядке?

Внешне все может выглядеть неплохо (рис. 1), но при первых же попытках внесения изменений вы увидите: что-то

отображению на экране; перейдя в новое качество, она как бы приобретает гибкость и всякий раз аккуратно укладывается программой в заданную ширину, в несколько эквивалентных строчек.

Другое неудобство – “форматирующие пустоты” Лексикона: добавочные пробелы и лишние строки. Поскольку MSW смотрит на оформление как на самостоятельный процесс, независимый от редактирования, эти пустоты программа не отличает от обычных строк и пробелов. Поэтому левое поле (и независимо от его присутствия – красная строка), а также добавочные строки, введенные при нумерации страниц, просто засоряют текст в MSW. Форматирующие пустоты обеспечиваются в MSW другими средствами.

Ситуацию с пустотами можно считать терпимой: ведь они удаляются как обычные символы. С концами строк дело хуже, потому что объем редактирования здесь гораздо больше: придется “сливать” группы соседних строк-псевдоабзацев по всему тексту.

К счастью, программисты, которые давно осознали проблему, приняли нужные меры. Специальная программа-преобразователь, **ABC-WORD** (автор А.Чижов), поможет нам справиться с трудностями. К ней следует обратиться перед загрузкой файла в MS Word. Программа возьмет обычный текстовый файл, проделает работу по распознаванию абзацев, которую мы выше пытались возложить на сам Word, и оставит концы строк только там, где необходимо. Придется, правда, немного ей помочь и поработать над текстом в Лексиконе.

В силу причин, о которых мы говорили выше, ABC-WORD распознает только регулярные абзацы. Признаком нового абзаца мы условились считать пробел в начале очередной строки. Поэтому из файла прежде всего нужно убрать левое поле, образованное символами пробела. (Это может оказаться не так просто. “Одним махом” уничтожить отступ от края не удастся, если поле неодинаково по всему тексту.) “Нормальная” красная строка как раз образована в Лексиконе несколькими начальными пробелами, здесь все в порядке, а с “нависающими” красными строками придется повозиться, либо... смириться с этим изъяном текста и заняться необходимым редактированием уже средствами Word.

Лучше всего, если файл, подготовленный в Лексиконе для преобразования, будет выглядеть следующим образом (рис. 3). Буква “а” обозначает начало “истинных” абзацев.

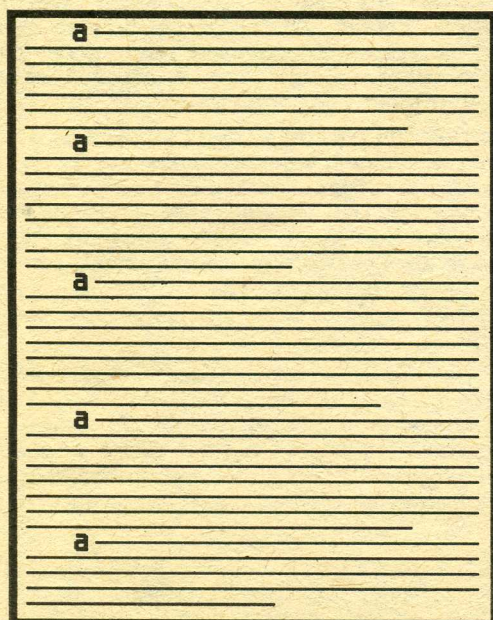


Рис. 3. Регулярный текст, “перевариваемый” программой ABC-WORD

В результате работы программы ABC-WORD, получится файл, который будет правильно прочитан в MS Word.

Таким образом, вся процедура преобразования текстового файла в формат Word выглядит следующим образом:

- 1) уберите в исходном файле левое поле;
 - 2) избавьтесь от “нависания”: красные строки должны быть втянуты;
 - 3) позаботьтесь об отступе в начале абзацев (один или больше пробелов); вместо начального пробела абзац может быть отделен от предыдущего пустой строкой;
 - 4) избавьтесь по возможности от всякого иного форматирования;
 - 5) обратитесь к программе преобразования текста ABC-WORD <FILENAME.LEX> FILENAME.DOC
- Здесь FILENAME.LEX – входной файл (из Лексикона), а FILENAME.DOC – выходной файл, готовый к загрузке в Word;
- 6) обратитесь к MS Word с именем вновь созданного файла (лучше всего – из каталога, содержащего данный файл): WORD FILENAME.DOC

“Физиономия” Microsoft Word

На экране – содержимое нашего файла. Оно заполняет окно, ограниченное рамкой. Это поле основной деятельности – ввода нового текста и внесения изменений. Курсор находится на первом знаке текста. (Возможно, что вдоль левой рамки внутри окна будут разбросаны звездочки; это зависит от начальных параметров системы. Звездочки расположены напротив каждого абзаца и лишний раз подтверждают правильность проведенного преобразования.)

Но поговорим сначала о других элементах экрана, находящихся за пределами рамки. Наиболее существенным и важным из них является командное меню, расположенное в двух строках на “подоконнике” (рис. 4); возможен режим, когда оно закрыто рамкой. На данном этапе знакомства мы должны воспринять наличие этого меню как информацию о том, что кроме прямого редактирования есть некоторая “косвенная” деятельность, состоящая в выполнении команд. Наиболее употребительные команды продублированы в виде “быстрых клавиш”, что позволяет выполнять их, не выходя из режима прямого редактирования. Однако в целом характер команд и их количество таковы, что сосредоточить их все на комбинациях клавиш невысказано. Отметим, что подобное меню (располагаемое в верхней части экрана) имеется и в Лексиконе, но там оно менее разветвленное.

объявлен значением поля записи; обеспечивается просмотр файла из Базы данных. Это позволяет использовать готовую информацию, ранее введенную в традиционном текстовом редакторе. ♦

COMMAND: Copy Delete Format Gallery Help Insert Jump Library
Options Print Quit Replace Search Transfer Undo Window
Copies selected text to scrap or to a named glossary entry
Pg1 L11 Col (Ap-18) ? ZH

TEST.DOC

Microsoft Word

Рис. 4. Меню команд MS Word на “подоконнике”

Нажав клавишу [Esc], мы перейдем от прямого редактирования к выполнению команд меню. Редактирование в этом режиме невозможно. (Та же клавиша [Esc] при необходимости возвратит нас в рамку.) Сигналом о том, что меню “активно”, является подсветка одной из его команд, в нашем случае первой в списке команды, *Copy* (команды расположены в алфавитном порядке).

О назначении команд мы, впервые оказавшиеся в мире Word, можем только догадываться. Некоторую дополнительную информацию дает строка расшифровки, расположенная ниже списка команд (прием, вполне типичный для меню). Воспользуемся случаем и обойдем весь список – важ-



ный "перекресток" MSW. Количество "дорог", ведущих отсюда, равно числу команд в списке (не считая возврата в рамку). Обход меню выполняется с помощью клавиши [Пробел] или курсорных стрелок. Обратимся за пояснением к строке расшифровки. При последовательном обходе нашей "площади" обнаружим, что вернулись к началу, т.е. список - круговой. Одну из команд нам под силу осознать и выполнить уже сейчас - это команда *Quit*, которая, согласно пояснению, "оканчивает сеанс работы с Word" (разумеется, все расшифровки, как и сами команды, даются на английском языке, если только вы не имеете дело с русифицированной версией). Выполнить команду можно двояко: первый способ состоит в том, что мы сначала высвечиваем команду, а затем, нажав [Enter], выполняем. Второй более быстрый способ выполнения - по первой букве команды. Итак, нажимаем [Q] - до свидания Word, здравствуй DOS! Заканчивать работу с программой тоже надо уметь...

Теперь повторно войдем в программу, но намеренно сделаем это не так, как в прошлый раз, а в упрощенном варианте:

WORD

На этот раз перед нами пустая рамка: файл не был указан, а потому и не загружен. Теперь, когда нас не отвлекает текст, рассмотрим более мелкие элементы "декора" экрана.

Левый верхний угол рамки содержит цифру 1 - номер окна. Этот факт наводит на мысль (совершенно верную), что окон может быть много. MSW позволяет одновременно работать с 8 окнами, в которых мы можем видеть и разные тексты, и различные части одного и того же документа. В отличие от других многооконных систем (в том числе от Лексикона) окна MSW не перекрываются. Каждое очередное окно создается вертикальным или горизонтальным делением одного из имеющихся. Они могли бы выглядеть так:

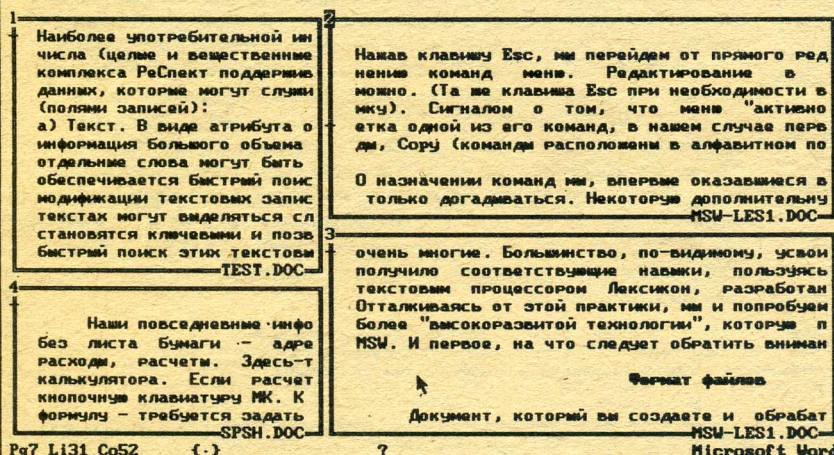


Рис. 5. Окна MS Word не пересекаются

С каждым текстом работа ведется только в пределах окна и конечно же работать "в узких рамках" неудобно. Поэтому предусмотрено "распахивание" во весь экран (укрупнение).

В этом случае окна, как бы стоят друг за другом (рис. 6), и просматривая их (F1), мы можем выбрать "нужные рамки".

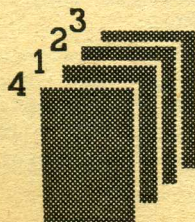


Рис. 6. При укрупнении во весь экран окна как бы стоят друг за другом

Несколько элементов содержит самая нижняя строка экрана (рис. 2): слева - номер страницы, с которой вы работаете (Word версии 5.0 автоматически размечает и нумерует страницы; в более ранних версиях нумерация производится по команде пользователя и виден лишь результат последней разметки), номер строки на странице (необязательный элемент) и номер позиции в строке.

Затем стоят скромные фигурные скобки, играющие, однако, важную практическую роль. Это "корзина", в которой собирается то, что мы удаляем из текста. Ее содержимое и изображается между фигурными скобками. (Если удаленный кусок текста велик, то показывается только его начало и конец, разделенные многоточием.) В основном это "обрезки": отдельные буквы, куски слов, но иногда целые фрагменты, которые вполне еще можно использовать. Дело в том, что "мусор" из корзины выбрасывается не сразу, а лишь при поступлении следующей порции. Однако до этого момента "отходы производства" еще могут быть "возвращены", т.е. вставлены в любое место текста.

Знак вопроса в середине нижней строки - "точка под сказкой"; она служит для вызова справочной системы (Help) при манипуляторе "мышь". В оставшейся части строки - условные значения некоторых рабочих параметров, а также название самой программы.

Закончив обзор элементов экрана, вернемся к нашему файлу. Чтобы приступить к редактированию, произведем его загрузку.

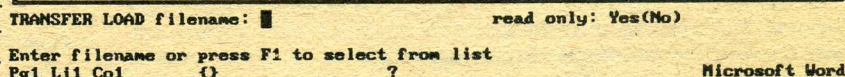


Рис. 7. Подчиненное меню: поворот в боковую улочку

Для этого:

- 1) перейдя в меню команд, нажмем клавишу [Esc];
- 2) выберем команду *Transfer* (Обмен с диском).

Мы увидим новое меню, отражающее варианты команды *Transfer*. Отвечаясь от основного, оно показывает, что общий принцип организации команд - древовидный (рис. 7). Таким образом, от основного меню мы можем двигаться в разных направлениях, "сворачивая" каждый раз на нужную команду. В отличие от реальной уличной сети никакие два направления не пересекаются: к каждому пункту ведет лишь один маршрут.

Выбрав теперь команду *Load* (Загрузить), мы увидим перечень полей, представляющих собой параметры команды. (Этим кончается всякий маршрут в системе команд; сам пройденный маршрут, кстати, показывается в левой части "подоконника".) Параметрам нужно придать те или иные конкретные значения. В данном случае у нас два поля: параметры "имя файла" и "только для чтения"; в других командах их может быть значительно больше.

Некоторые параметры имеют небольшой набор возможных значений; в этом случае все они выведены на экран рядом с параметром, а одно из них предопределено - заключено на экране в круглые скобки. Такие параметры можно не устанавливать, и тогда система использует дежурное значение. В нашем случае таков параметр

read only: Yes(No),

означающий: "документ будет не только прочитан, но, возможно, изменен". (Выбор другого значения позволил бы нам просматривать документ, но не менять его.) Параметр

filename:

другого типа; его значение нам придется сообщить системе, никакого дежурного имени файла не существует.

Чтобы и здесь облегчить выбор, система может предложить подсказку. Нажав "подсказывающую" клавишу [F1], получим перечень файлов в текущем каталоге ДОС (см. рис. 8), из которого выберем свой файл. Подтвердив выбор клавишей [Enter], завершим выполнение команды *Transfer Load* (Загрузка файла).

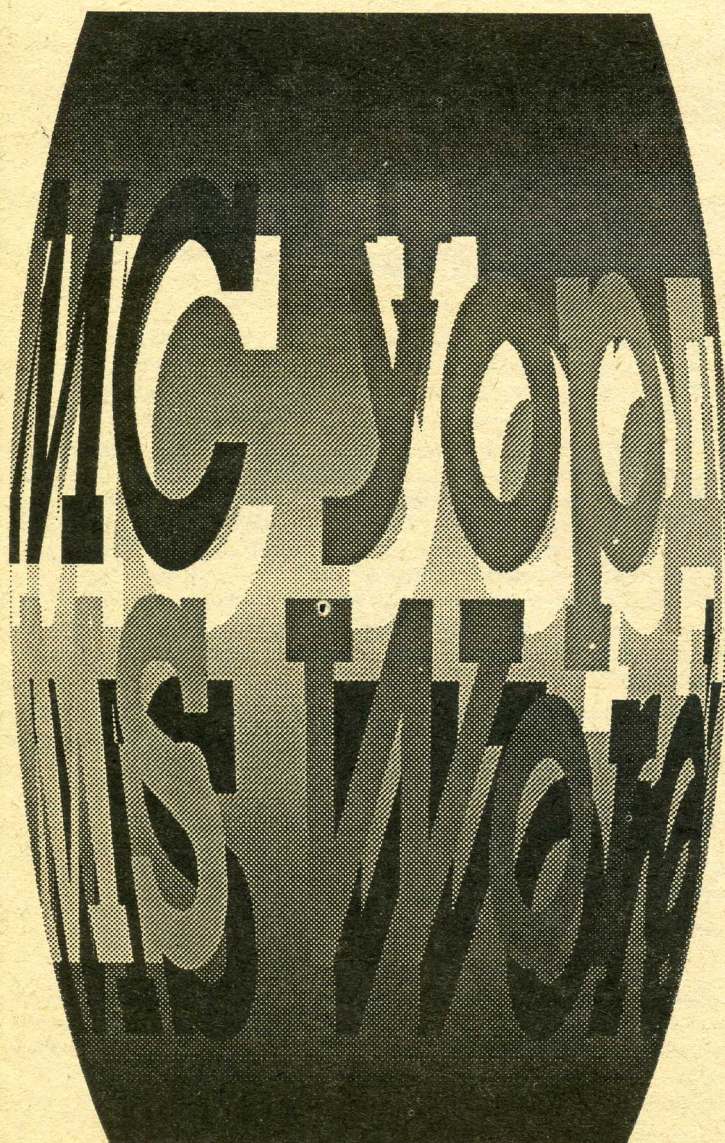
```
C:\WORD\*.DOC
MSW-LES1.DOC [B:]
TEST.DOC [A:]
```

```
TRANSFER LOAD filename: MSW-LES1.DOC read only: Yes(No)
Enter filename or press F1 to select from list (18884352 bytes free)
```

Рис. 8. Подсказка имени файла

Мы возвращаемся в режим редактирования, в окно выведен уже знакомый текст нашего файла. Добавился еще один элемент декора окна: имя файла на нижней рамке справа.

Теперь можно приступить непосредственно к редактированию. Как мы позже увидим, оно имеет много общего с редактированием в Лексиконе. "Общие места" процесса редактирования помогут нам быстрее освоить MS Word. Но об этом — в следующей статье.



компьютер на работе

Переписка с читателями

После публикации во втором выпуске сборника статьи о программах-упаковщиках мы получили несколько читательских откликов. Предлагаем вашему вниманию наиболее интересные из них.

Т. Цыганко из г. Харькова сообщает, что преимущества программы Lharc более всего проявляются при упаковке большого количества маленьких файлов и при упаковке плохо сжимаемых файлов. В частности, Lharc иногда может упаковывать даже архивы, созданные Pkzip и Pkarc (обратного не наблюдалось). Он отмечает также, что свойство Lharc упаковывать файлы в алфавитном порядке не всегда удобно, в частности при упаковке с подкаталогами.

А. Румянцев из г. Калининграда Московской обл. проводил тестирование программ Pkarc, Pkzip, Pak и Lharc. К сожалению, он не указал версии этих программ и конфигурацию компьютера, на котором производилось тестирование. Полученные им выводы таковы:

- 1) программы Pkzip и Lharc сжимают файлы на 7-8% эффективнее по сравнению с Pkarc;
- 2) программа Pak является промежуточной по всем параметрам;
- 3) текстовые файлы все упаковщики сжимают практически одинаково, поэтому для них лучше использовать самый быстрый — Pkarc;
- 4) при упаковке выполняемых файлов лучше всего использовать программу Pkzip;
- 5) программа Lharc сделана очень профессионально, но имеет два недостатка: низкую скорость работы и отсутствие защиты архивных файлов с помощью пароля.

В целом, по мнению А. Румянцева, наилучшей программой-упаковщиком является Pkzip, но на машинах типа XT можно пользоваться и Pkarc в сочетании с программой-оболочкой Narg.

В. Фигурнов из г. Москвы предоставил нам свои результаты тестирования последних версий программ-упаковщиков. Сравнивались следующие программы:

- Pkzip/Pkunzip фирмы PKWare, версия 1.10;
- Pak фирмы NoGate Consulting, версия 2.10;
- Ice (Lharc), разработанная Nagayasu Yoshizaki, версия 1.14.

Программа Pkzip тестировалась в двух режимах: обычном, обеспечивающим максимальное сжатие, и с параметром -ES, при котором достигается быстрая работа программы при несколько худшем сжатии. Для тестирования были выбраны четыре разнородные группы файлов:





- шрифты для лазерного принтера (далее - шрифты);
- редактор текстов Microsoft Word 5.0 (далее - Word);
- тексты программ на языке Pascal из библиотеки Turbo Professional 5.5 (далее - тексты программы);
- документы, подготовленные в Microsoft Word (далее - документы).

В каждую из этих групп входило 20-30 файлов общим объемом около 1 Мбайта. Тестирование проводилось на компьютере IBM PC AT (12 МГц) с дисководом Seagate ST-251. Результаты тестирования оказались следующими (лучшие показатели выделены полужирным шрифтом).

Таблица 1

Степень сжатия, обеспечиваемая программами-упаковщиками (отношение размера архива к общему объему исходных файлов)

	Pkzip	Pak	Ice	Pkzip -ES
Шрифты	37.7%	37.4%	37.3%	45.9%
Word	64.4%	65.6%	65.3%	79.4%
Тексты программ	25.8%	26.6%	27.3%	37.8%
Документы	38.6%	36.9%	38.1%	46.3%

Таблица 2

Время упаковки 1 Мбайта файлов, с

	Pkzip	Pak	Ice	Pkzip -ES
Шрифты	211.5	189.9	203.6	36.6
Word	110.8	162.7	222.5	55.4
Тексты программ	95.8	131.6	194.5	32.9
Документы	149.6	131.3	207.6	36.6

Таблица 3

Время распаковки 1 Мбайта файлов, с

	Pkzip	Pak	Ice	Pkzip -ES
Шрифты	31.9	30.8	91.0	40.2
Word	31.2	54.8	132.0	38.6
Тексты программ	33.8	48.8	83.6	40.4
Документы	24.4	40.3	76.9	31.8

Таким образом, ни одна из программ не является лучшей по всем показателям. Даже если сравнивать их только по обеспечиваемой степени сжатия при помещении файлов в архив, то иногда лучшие результаты дает одна программа, а иногда - другая. Напомним, что такая же картина наблюдалась и для предыдущих версий этих программ (см. выпуск 2). Практика показывает, что в среднем программа Ice (Lharc) дает на 5-7% лучшее сжатие. Однако эта программа и самая медленная - порой она требует времени в 2-4 раза больше, чем другие.

Быстрее всего помещает файлы в архив программа Pkzip с параметром -ES, однако она сжимает файлы на 20-30% хуже других. Если такая плата за скорость работы не является приемлемой, то следует предпочесть Pkzip без параметра -ES. По скорости распаковки файлов вне конкуренции Pkzip.

Важное преимущество программы Pkzip по сравнению с Pak и Ice состоит в том, что она хранит в архивном файле всю информацию об оплавлении архива в двух экземплярах, что позволяет с помощью программы Pkzip'ix восстанавливать информацию из поврежденного архивного файла с минимальными потерями.

Таким образом, если обеспечиваемая программой степень сжатия файлов настолько важна, что ради нее можно пойти на замедление скорости работы и снижение устойчивости архивных файлов к повреждениям, то надо использовать программу Ice. В остальных случаях лучше пользоваться Pkzip/Pkzip'ix. В.Фигурнов приводит также таблицу оценок отдельных свойств программ архивации по пятибалльной шкале:

	Pkzip	Pak	Ice	Pkzip -ES
Степень сжатия	4	4	4-5	3
Скорость упаковки	4	3-4	2	5
Скорость распаковки	5	4	2	4-5
Восстановление сбойных архивных файлов	4	2	2	4
Интерфейс	3	3	3	3

Русификация Norton Commander

В первых двух выпусках сборника за этот год были опубликованы статьи про Norton Commander. Мы получили несколько писем с советами, как можно его "русифицировать". Приводим рекомендации, полученные от В. Фигурнова. (Аналогичные советы мы получили от А. Тереньева из г. Москвы.)

В своем оригинальном виде Norton Commander 3.0 "не воспринимает" русскую букву "р". Указанный недостаток может быть устранен заменой в файле NCMAIN.EXE байта с шестнадцатеричным смещением 1E02. Значение этого байта надо изменить с шестнадцатеричного E0 на 00. Такое исправление можно выполнить, например, с помощью программы NU.

Аналогичные замены следует произвести и в других программах NC - WPVIEW.EXE, DBVIEW.EXE и т.д. Найти байты, которые надо исправить, можно следующим образом. Причиной "неправильного" поведения Norton Commander является фрагмент программы обработки ввода с клавиатуры:

```
cmp byte ptr [bp-2], 000h
je    *+1bh
cmp byte ptr [bp-2], 0e0h
je    *+15h
```

Этот фрагмент в теле программы представлен байтами:

80 7E FE 00 74 1B 80 7E FE E0 74 15

Для того чтобы Norton Commander "воспринимал" букву "р", следует с помощью NU найти во всех его программах данную последовательность байтов и заменить в ней значение третьего байта от конца с E0 на 00.

Кроме того, Norton Commander имеет таблицы перевода символов с кодами 128-165 (содержащих в кодировке IBM буквы европейских алфавитов) из прописных в строчные и наоборот (они применяются, например, при поиске строк в файлах). Данные таблицы можно скорректировать для использования Norton Commander с русскими текстами. Это не полностью адаптирует Norton Commander для работы с русскими буквами, поскольку они располагаются и вне диапазона кодов 128-165, но все же такая корректировка полезна.

Для корректировки таблиц надо сделать следующие исправления в файле NCMAIN.EXE (значения всех смещений даются в шестнадцатеричном виде):

- байты 21902-21921 - буквы а-я (кроме Я);
- байты 21922-21927 - буквы а-е;
- байты 21928-21947 - буквы А-Я (кроме П);
- байты 21948-2194D - буквы А-Е.

Перед заменой посмотрите, что на этом месте было раньше, и сделайте аналогичную замену в других программах Norton Commander.

\компьютер на работе\

© Дональд Качоровский

OPEN ACCESS II

Что такое хороший пакет прикладных программ общего назначения? Это программная система, которая может помочь вам во всех случаях жизни. Может быть, за исключением момента, когда надо сварить кофе.

Судя по всему, к таким универсальным пакетам следует причислить **Интегрированную систему "OPEN ACCESS II"** (OA), предназначенную для IBM-совместимых компьютеров. Система является продуктом фирмы Software Products International (США), а ее полную русификацию выполнило СП "Интерсофт", образованное Институтом проблем информатики АН СССР и Software Products International (SPI). "Интерсофт" занимается также продажей OAI и обслуживанием пользователей в СССР. Система OAI с успехом продается во многих странах, а в соответствии с политикой фирмы SPI непрерывно ведутся работы по созданию национальных версий пакета. Русская языковая версия стала уже двенадцатой.

OA может стать незаменимым помощником делового человека, руководящего малым или средним предприятием. Система позволяет продуманно организовать рабочее время и в любой момент воспользоваться необходимыми инструментами для работы с базой данных, обработки текста, выполнения вычислений и связи с внешним миром.

Пакет OPEN ACCESS II состоит из восьми дискет, в том числе двух с примерами для изучения работы системы, и семи книг. На дискетах записаны модули пакета, а книги являются руководствами по использованию системы. Первым делом надо ознакомиться с самой тонкой из книг, озаглавленной "Первое знакомство". В ней вы найдете информацию о том, как установить OA на вашем компьютере. Система подготовлена таким образом, что установку может выполнить даже тот, кто практически ничего не знает о компьютерах. Единственное предварительное требование - это наличие в памяти машины драйвера русского экранного шрифта в альтернативной кодировке.

После первоначальной установки можно немедленно начать работу. Пользователь получает доступ к следующим элементам системы: "База данных", "Электронная таблица", "Текстовый редактор", "Программер", а также "Рабочий стол" и "Сервис".

Названия первых трех модулей говорят сами за себя и можно только добавить, что обмен данными между модулями проходит легко и быстро, а их возможности настолько широки, что, судя по всему, могут удовлетворить самые разнообразные потребности пользователей, нуждающихся в программных инструментах, ориентированных на базу данных.

Что касается "Программера", то это достаточно простой язык программирования интерпретирующего типа, позволяющий создавать собственные прикладные программы, включенные в систему OA (например, имеющие доступ к базе данных). Разработчики утверждают, что на этом языке можно написать как программу составления платежной ведомости или контроля продукции, так и игру (даже менеджер должен иногда отдыхать).

Чрезвычайно удобным модулем системы Open Access является "Рабочий стол". Он имеет целый ряд функций, которые можно назвать "электронной оргтехникой". Вы можете запланировать расписание занятий на каждый день, а система напомнит вам о некоторых из них звуковым сигналом и соответствующим сообщением; в любой момент есть доступ к календарю, часам и записной книжке; имеется картотека "визитных карточек" ваших знакомых и деловых партнеров (если к компьютеру подключен модем, то система может набрать телефонный номер из указанной карточки); на "Рабочем столе" всегда под рукой калькулятор, способный выполнять даже сложные статистические вычисления; есть дополнительный текстовый редактор (не всегда стоит использовать мощный основной модуль обработки текстов); можно использовать функцию "Преобразование", которая позволяет перевести значения из одних единиц измерения в другие. И еще много других полезных функций.

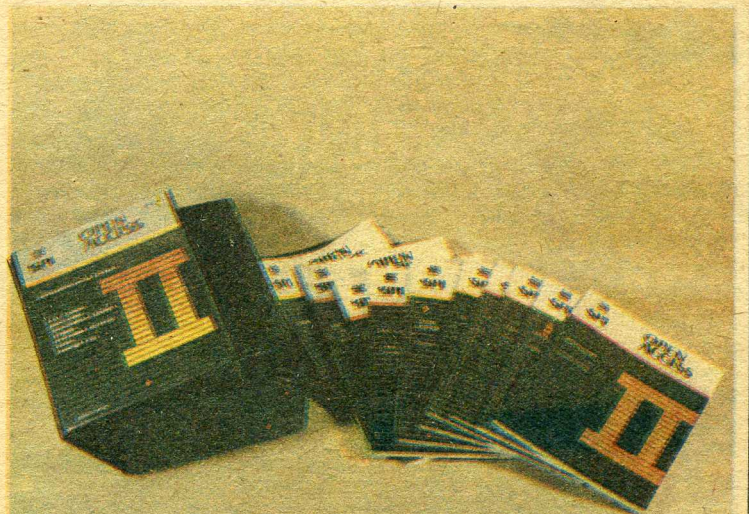
Важную роль в ежедневном использовании играет также модуль "Сервис", являющийся своего рода "техническим обеспечением" системы OA. "Сервис" предоставляет широкие возможности конфигурирования системы как в аппаратном отношении (установка принтера, плоттера, задание устройств для поиска данных и т.д.), так и в аспекте связи OA с компьютером. Второе важное свойство "Сервиса" - это возможность преобразования формата данных, как с целью обмена данными между различными модулями OPEN ACCESS, так и для передачи данных между OA и другими системами (dBaseII, dBaseIII, Lotus и т.д.).

И это отнюдь не все возможности Интегрированной системы OPEN ACCESS II. В качестве дополнительных поставок предусмотрены такие модули, как "Статистика" и "Коммуникации" (для использования последнего необходим модем). Для пользователей, заинтересованных в создании собственных приложений на языке "Программер", разработан компилятор этого языка. Существует также сетевая версия системы OA.

Практически в любой части документации к системе OA (в конце каждого пособия) пользователь найдет таблицу функциональных команд, применяемых в данном модуле. Сводная таблица находится в "Первом знакомстве". Расхождений между командами в разных модулях нет, т.е. если вы привыкнете к командам текстового редактора, то не будете делать ошибок, работая с электронной таблицей или базой данных.

Сегодня мы познакомили вас только с основными элементами и возможностями OPEN ACCESS II. В одном из следующих выпусков будет опубликовано более подробное описание этой системы.

Перевод Халины Мадейчик



Компьютер на работе

© Тимур Цыганко

НОВИНКИ FIDO

Этой статьей мы начинаем обзор программ, доступных в Kremlin FIDO (см. "Компьютер" N 1 и 2).

4DOS - это операционная "оболочка DOS", но она совершенно не похожа на большинство других оболочек. Такие популярные и распространенные системы, как Norton Commander, XTreePro, PC Shell и др., отдают пользователю от команд DOS и предоставляют свои средства и методы работы, в то время как 4DOS предлагает существенно облегченную и значительно более эффективную работу в режиме "командной строки".

Каковы же достоинства 4DOS? Это - более 40 новых и значительно расширенные возможности традиционных команд (например, команда DIR имеет 23(!) ключа), условное выполнение команд, несколько команд в строке, сохранение, редактирование и использование ранее введенных команд. Это - работа в режимах EGA и VGA, поддержка MS-DOS совместимых сетей, включая 3Com 3+ и Novell Netware. Пакетные файлы в 4DOS скорее напоминают Бейсик и выполняются очень быстро. 4DOS снабжен документацией, а также встроенной подсказкой по своим командам и командам MS-DOS.

4DOS может быть запущена как обычная программа либо может подключаться вместо COMMAND.COM путем указания в строке SHELL файла CONFIG.SYS. Размер резидентной части 4DOS - не более 4 Кбайт. Для пользователей с 80286 и выше имеется оптимизированная версия.

Из почти любого EXE-файла программа **LZEXE** делает тоже EXE-файл, но гораздо меньшего размера. Идея проста - исходный файл упаковывается, а при запуске автоматически распаковывается. Процент сжатия реальных программ колеблется от 25-30% (GW Basic) до 50-60% (Norton Utility).

Нельзя упаковать только программы со встроенными оверлеями, но, обнаружив их, программа сама предупредит вас об этом. И пожалуй, не стоит сжимать антивирусные программы - обычно появляется сообщение о повреждении такой программы.

Если вам нужно уменьшить COM-файл, воспользуйтесь утилитой COMTOEXE и вы преобразуете программу в формате COM к виду EXE. Однако не рекомендуется пытаться упаковывать COMMAND.COM. Даже явное указание SHELL в файле CONFIG.SYS и SET COMSPEC в AUTOEXEC.BAT не спасут вас от ошибок распределения памяти.

Какова же цена высвобождения дискового пространства? Около 1 секунды времени при загрузке на AT/12 МГц упакованной программы объемом 200 Кбайт. А при загрузке программ с дискета вы даже выиграете по времени, так как короткий файл и прочитается быстрее.

INFOPLUS - это, собственно, программа определения некоторых параметров компьютера и DOS. Но самое главное - это прекрасный образец для любителей прямого доступа к оборудованию и системе из языков высокого уровня. Внимательное изучение исходных текстов этой про-

граммы (19 модулей на Turbo Pascal и 1 на Turbo Assembler) облегчит жизнь любому, кто намерен выйти за пределы возможностей DOS.

С помощью INFOPLUS можно получить информацию о: компьютере и ПЗУ; системных программах; процессоре и сопроцессоре; переменных среды (environment); оперативной памяти; распределении блоков памяти; загруженных драйверах; устройстве на уровне DOS и BIOS; видеоадаптере; клавиатуре и мышке; параллельных/последовательных портах; дисках; таблице 파티ций; DOS.

Что делать, если программа требует расширенной памяти (EMS), а у вас ее нет? Как использовать, кроме организации виртуального диска, увеличенную память (extended memory)? Ответ прост - воспользуйтесь **ABOVE DISC**. Эта уникальная программа позволяет эмулировать EMS в увеличенной памяти или на винчестере, причем работать она может даже на компьютере с 8088 процессором!

При установке программы необходимо указать:

- количество страниц эмулируемой EMS;
- выполняется ли эмуляция в увеличенной памяти или на диске (каком);
- будет ли область отображения страниц расположена в основной памяти, в основной памяти на границе 16 Кбайт, в памяти с указанного сегмента или будут использованы старшие 64 Кбайта основной памяти для MS-Windows;
- режим Normal/80286/80386;
- используемую шину - Standard Bus/Micro Channel.

Необходимо отметить, что некоторые программы, например редактор MultiEdit, работают нормально только при использовании для отображения страниц старших 64 Кбайт памяти. С другой стороны, в этом режиме драйвер всегда забирает 64 Кбайта сразу при загрузке, независимо от того, будет ли применяться эмулятор или нет.

И последнее. В данной версии программы не работает поддержка 80386, но можно работать в режиме Normal.

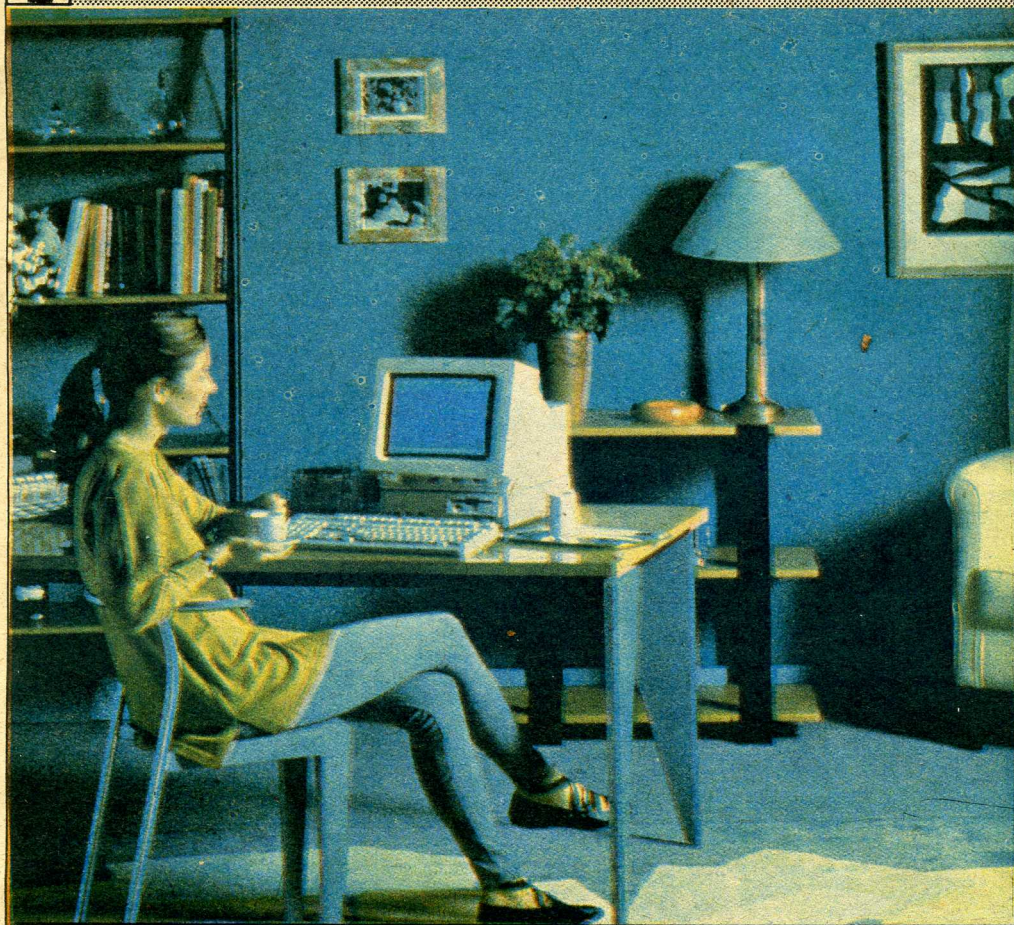
Файл **SLOWER.ZIP** содержит программы AT! и SLOWER, которые замедляют работу компьютера. Если вы хотите поиграть в старый добрый Digger на PC/AT, вам не обойтись без этих программ. Преимущество AT! в возможности изменять скорость "на ходу" - нажав клавиши [Правый Shift] + [Серый Минус] или [Правый Shift] + [Серый Плюс].

В файле **KILLER.ZIP** - программы PK_XT и PK_AT, обеспечивающие мгновенный выход в DOS, при котором уничтожаются все загруженные после них резидентные программы. Эти программы отличаются используемыми клавишами: для XT это [PrtScr] и [ScrLock], для AT - [F11] и [F12].

Прилагается список советских и "почти советских" BBS

NAME	TOWN	NUMBER
FIDO	Mscw	8W(095)205-3554
VNIИ BAS - Xenix!	Mscw	8W(095)329-3744
Morning Star	Novosib	8W(383)235-67-22
SVP BBS	Novosib	8W(383)235-45-70
MamBox BBS new	Taln	8W(0142)443-360
Hackers Night Sys.1	Taln	8W(0142)442-143
Hackers Night Sys.2	Taln	8W(0142)601-818
Micro BBS	Taln	8W(0142)444-644
P.O. Box Maximus	Taln	8W(0142)529-237
Eston. BBS.1 Eesti	Taln	8W(0142)422-583
MESO BBS	Tart	8W(014)343-3434
Goodwin BBS	Taln	8W(0142)691872
Hacker's Inn BBS	Taln	8W(0142)423178
Mail Shark BBS	Taln	8W(0142)532350
PaPeR BBS	Taln	8W(0143)433351
Lion's Cave BBS	Taln	8W(0142)536246

И. Свиридов (044) 263-8770 Voice & Data 24 Hour



Компьютер дома

-  Как сделать "отмычку"
-  Тайны TR-DOS
-  Эмулятор ПЗУ для ZX Spectrum
-  Спектрум и принтер
-  Centronics для ATARI XL/XE
-  Русские буквы на ATARI ST
-  First XLEnt
-  Микро-микро-компьютер
-  Какой Бейсик выбрать?
-  Рисуем с ЭВМ
-  Музыкальная шкатулка



\компьютер дома\

© Гжегож Чапкевич

Как сделать "отмычку"

На компьютерном жаргоне "взломщиком" называют программиста, который декодирует шифры и "пароли" банковских компьютерных систем. Мы хотим предложить вам стать "взломщиками" другого типа. "Взламывать" программы можно не для того, чтобы незаконным путем облегчить себе жизнь, а чтобы упростить игру, меняя ее параметры. Предупреждаем: это увлекает и быстро становится самым интересным и самым важным элементом контакта с компьютером. Результатом может стать знакомство с языком Ассемблера и разными приемами программирования, которые впоследствии пригодятся при работе с ЭВМ.

На примере нескольких относительно старых и распространенных программ для ZX Spectrum мы постараемся показать, как обеспечить себе "бессмертие", получить неограниченное количество "снарядов" или отключить счет-

чик ошибок. Если вы знаете язык Ассемблера микропроцессора Z80, то понять примеры будет легко, если не знаете - заглядывайте по ходу дела на стр. 35, где опубликован список команд этого микропроцессора.

Прежде всего давайте договоримся, что регистры A, B, HL будут рассматриваться как... переменные (что не столь далеко от истины). Другим новым элементом, с которым вы не встречались в Бейсике, будут шестнадцатеричные числа. Основой системы является 16, а цифрами, кроме хорошо знакомых 0, 1, 2, 3 и т.д., - также A, B, C, D, E и F, которые соответствуют по значению десятичным 10, 11, 12, 13, 14, 15. В примерах числа и адреса будут записаны в шестнадцатеричной системе. Кроме того, запись (A26F) означает содержимое ячейки памяти с адресом A26F. Аналогично интерпретируется (HL) - адресом является значение регистра (переменной) HL.

Первым делом надо определить длину программы и ее стартовый адрес. В этом нам поможет программа COPY-COPY (см. "Компьютер" N 1, с. 57) позволяющая считывать программу, менять в ней заголовки и делить ее на части, поскольку в памяти должен еще поместиться монитор (например, MONS3M21 фирмы HiSoft). В настоящее время пишутся довольно длинные программы, поэтому редко когда в памяти помещается и целая программа, и монитор. Определение стартового адреса также перестало быть простой задачей.

Начать следует с вывода на экран первого блока, который чаще всего написан на Бейсике. Команды BREAK и MERGE обычно не помогают, так как программы их блокируют. COPY-COPY без труда обходит это препятствие. Командой LET измените параметр START в заголовке на десятичную величину 32768 и запишите измененный таким об-





разом заголовок на кассету. Теперь надо загрузить новый заголовок, а затем - остальную часть файла. Компьютер выдаст "OK" и можно просмотреть текст программы. Но радоваться рано. Нередко случается так, что экран остается пустым. Нужно снова воспользоваться `CORU-CORU` (это настолько мощный инструмент, что даже жалко пользоваться им для простого копирования программ). Имея `CORU-CORU` в памяти компьютера, загрузите "обрабатываемую" программу и воспользуйтесь командой `LIST`. Как известно, два первых байта каждой строки программы на Бейсике содержат ее номер (помните, что это единственный случай, когда первый байт является старшим). Может оказаться, что все строки имеют номер 0 или же номера расположены не в порядке возрастания. С помощью команды `POKE` надо приписать им соответствующие номера. Следующие два байта означают длину строки (для блокировки `MERGE` эта величина иногда записана неверно), а дальше уже идет текст. В нем надо отыскать десятичные числа 16 и 17, после которых находятся величины, описывающие цвет фона и символов ("чернил"). Командой `POKE` после каждого числа 16 впишите 0 (черная печать), а после каждого числа 17 - величину 7 (белый фон). Каждая строка программы на Бейсике заканчивается величиной 13, которая означает перевод строки - `ENTER`.

При определенной практике и имея под рукой команды Бейсика с помощью `CORU-CORU` можно расшифровать интересующую вас программу. Это полезно, так как на изменении номеров строк и цветов перечень "обманов" не кончается. Перед нами - основная сложность, т.е. изменение значений чисел. На Бейсике `ZX Spectrum` числа записываются двойкой: сначала идет их цифровое представление, а потом число 14 (десятичное) и пять байт двоичного представления. Эти-то пять байт и определяют значение числа, независимо от его цифрового представления, что позволяет ввести числа с другими значениями, чем те, которые видны на распечатке. Необходимо тщательно проверить все величины в программе и в случае необходимости изменить цифровые представления. Теперь "исправленную" программу запишите на кассету (помните об изменении заголовка). После устранивания из памяти программы `CORU-CORU` (например, командой `USR` без параметра, т.е. клавишей `U` и `ENTER`) загрузите свою программу и начните поиски стартового адреса.

В старых программах после команд типа `LOAD` находилась команда `RANDOMIZE` `USR` и адрес, который указывал начало процедуры в машинном коде. Теперь часто приходится искать начало программы на стеке.

Однако прежде чем вы начнете читать текст программы, присмотритесь внимательно к игре. Обратите внимание на то, как она начинается, какие надписи появляются в конце и каковы ее параметры. Другими словами, сколько можно сделать ошибок, есть ли ограничение времени и какие клавиши используются для управления.

Предположим, что стартовый адрес уже известен, а в память компьютера загружен текст программы и программа-монитор. Для начала возьмем игру `ALIEN 8`, в которой есть ограничение времени, можно допустить 5 ошибок и надо собрать и отнести в назначенное место 24 предмета. Такой информации должно быть достаточно для достижения "бессмертия".

6300	DI	старт программы
	LD SP, F100	назначение показателя стека
	JP A631	переход под адрес A631

Просматривая программу с адреса A631, вы находите команды:

A65A	LD A, 05	присваивание A величины 5
	LD (5B1A), A	пересылка в ячейку 5B1A

Z80 имеет мало переменных (регистров) и большая часть информации хранится в соответствующих адресах памяти. Величина 5 выглядит "подозрительно", тем са-

мым "подозрительной" становится ячейка памяти с адресом 5B1A. В ходе игры программа будет использовать ее, чтобы проверить, исчерпано ли уже количество допустимых ошибок. Если нет, то счетчик ошибок будет уменьшаться на 1 и можно будет играть дальше, в противном случае программа возвратится к началу игры.

Поищите в памяти места, где использован адрес 5B1A. Программа `MONS3` делает это автоматически с помощью команды "G" (найди заданную последовательность). Команда "N" продолжает поиски. Если того, что вы ищете, нет в программе, то оно может быть в пространстве, которое занимает `MONS3`. Об этом надо помнить!

В этом случае вы ищете последовательность 1A, 5B, так как 2-байтовые адреса записываются в обратном порядке: сначала младший, затем старший байт.

Итак, вы находите:

CA15	LD HL, 5B1A	подстановка в HL
CA18	DEC (HL)	уменьшение на 1 содержимого ячейки памяти, указанной регистром HL
		переход, если результат меньше 0 (M - минус)
	JP M, B761	возврат (RETURN)
	RET	

Подозрение превращается в уверенность, когда вы убедитесь, что с адреса B761 начинается процедура конца программы. Помните также, что в программе есть счетчик времени. Когда он достигает величины "0000", игра заканчивается. Далее надо будет использовать тот факт, что адрес B761 означает конец программы. Итак, поищите другие места, где есть пары B1, 67.

ADA2	LD HL, 5B36	на этот раз HL указывает на другую ячейку памяти
	LD B, 04	регистр B играет здесь роль переменной цикла
	XOR A	результатом является обнуление A
ADA8	OR (HL)	логическая сумма A и содержимого ячейки, на которую указывает HL
	INC HL	увеличение HL на 1
	DJ NZ ADA8	уменьшение на 1 регистра B и переход к ADA8, если B < 0, то цикл FOR B=4 TO 1 STEP -1
	JP Z B761	переход к B761, если результатом логической суммы содержимого четырех последовательных ячеек памяти является 0 (Z=0)
	RET	

Вы узнали, что четыре последовательные ячейки памяти, начиная с 5B36, содержат счетчик времени. Кроме того, переход к процедуре B761 происходит в первом случае, если A=FF (или десятично 255, ибо таков в арифметике микропроцессора Z80 результат вычитания 0-1), а во втором, если A=0. Это важно, так как процедура B761 вызывается также из другого места.

B038	LD HL, 5B40	в A записывается содержимое ячейки памяти, указанной регистром HL
	LD A, (HL)	к содержимому A прибавляется 1
	ADD A, 01	содержимое A переводится в десятичный вид
	DAA	A пересылается обратно в десятичной форме
	LD (HL), A	сравнение A с числом 24 (десятичным)
	CP 24	условный переход. Если результат < 0 (NZ - не ноль)
	JR NZ B04C	
	LD A, 01	
	LD (5B23), A	A записывается в ячейку 5B23
	JP B761	переход к B761
B04C	CALL A7A3	

Ячейка с адресом 5B40 содержит количество собранных предметов, и если собраны все 24, то наступает переход к процедуре B761, но с A=1, что означает успешное завершение игры.

Теперь вы располагаете информацией, позволяющей ввести поправки. В ячейку с адресом A65B можно вписать величину, отличающуюся от 5, тем самым изменяя допустимое количество ошибок. В ячейку с адресом CA18 можно ввести 0, что означает NOP (нет операции) - вместо уменьшения содержимого ячейки 5B1A эффектом будет "бессмертие". Таким же образом вместо величины 24 можно записать другую, например 1, и тогда программа "поздравит" вас уже после выполнения первого задания.

Программы фирмы ULTIMATE имеют аналогичную структуру и программа KNIGHT LORE отличается от предыдущей лишь количеством предметов, которые надо собрать, и, конечно, адресами:

AF92 LD A, 05
LD (5BBA), A

и далее:

D13C LD HL, 5BBA
DEC (HL)
JP M BA22

Такой метод наиболее удобен - память расходуется экономно, так как само уменьшение счетчика ошибок занимает только 4 байта. KOKOTONI WOLF фирмы Elite использует для этого 7 байт:

A4E5 LD A, 06
LD (AEEA), A

Начало такое же, но дальше уже по-другому:

AADB LD A, (AEEA)	в A записывается содержимое
DEC A	ячейки с адресом AEEA
LD (AEEA), A	уменьшение содержимого A на 1
	A пересылается обратно в AEEA

Подобный метод уменьшения счетчика вы найдете в ALIEN HIGHWAY и BOOTU. Рассмотрим второй пример:

CD86 LD A, 04
LD (5BF1), A

и далее

E3B0 CALL 03B5	вызов процедуры BEEP из ПЗУ
LD A, (5BF1)	
DEC A	
LD (5BF1), A	
CP FF	проверка, будет ли результат
	вычитания A-1 меньше 0

Процедура 03B5 (Beeper) активизирует динамик компьютера ZX Spectrum. Стоит об этом помнить при поисках пути к "бессмертию" в других играх. А сейчас - еще два примера уменьшения счетчика ошибок. Первый, из программы BATMAN, лишь незначительно отличается от предыдущих:

8FB9 LD HL, 8F01	
LD A, (HL)	
SUB 01	вычитание 1 из величины
	содержимого A
DAA	
LD (HL), A	

Самый странный метод я встретил в программе ABU SIMBEL - PROFANATION. Процедура занимает целых 16 байт. Объявление предельно допустимого количества ошибок также решено нетипично:

C089 LD HL, 000A
LD (97A2), HL

здесь A означает шестнадцатеричное число

Иначе выглядит также уменьшение счетчика:

BA3E LD HL, (97A2)
PUSH DE
EX DE, HL

в стеке запоминается DE обмен значениями между HL и DE, т.е. значение ячейки 97A2 переходит в DE

LD HL, 0001
EX DE, HL

в HL снова попадает содержимое ячейки 97A2, DE содержит 1

XOR A

обнуление A с одновременной установкой флага C на 0

SBC HL, DE

вычитание из HL величины DE, соответствующее LET HL=HL-DE

POP DE

из стека берется прежнее значение DE

LD (97A2), HL

результат, наконец, попадает на место.

Довольно странный и запутанный способ. Похоже на то, что его единственная цель - ввести в заблуждение "взломщика".

Приведенные примеры лишь незначительно приблизили нас к "бессмертию". Главная трудность заключается в том, чтобы найти адрес ячейки, в которой хранится счетчик ошибок. Большинство программ в самом начале устанавливает стабильные параметры, которые после столь частого "GAME OVER" позволяют предпринимать очередную попытку. Однако не всегда можно с уверенностью сказать, что найденный адрес - это именно тот, который нам нужен. Программисты стараются ввести "взломщиков" в заблуждение и применяют неясные методы задания основных значений. Поэтому, если чтение текста программы не дает результатов, надо поискать "черный ход".

Найдите в программе надпись, информирующую о конце игры. Чаще всего это "GAME OVER", однако всегда сначала надо просмотреть игру и запомнить, чем она заканчивается. Например, программа "THREE WEEKS IN PARADISE" после поражения заполняет экран текстом, начинающимся словом "DEAD". С помощью функции G программы MONS3 отыщите последовательность чисел, соответствующих ASCII-кодам букв D, E, A, D. (Внимание: следует вводить значения кодов в шестнадцатеричной форме!) Найдите адрес BB21, с которого начинается текст, завершающий игру. Поиски этого адреса в тексте программы безрезультатны. Не остается ничего другого, как проверить соседние ячейки. В ячейке с адресом BB1D находится команда JP BAD1, занимающая три байта, дальше, в BB20, величина D0, а еще дальше, в BB21, начало слова DEAD. Если предположить, что величина D0 каким-то образом связана с текстом (например, определяет место первой буквы на экране), то следует отыскать адрес BB20. Вы найдете такую последовательность команд:

BB90 CALL A4C9
LD HL, BB20
LD DE, 0003

искомый адрес
здесь 3 определяет величину букв

CALL A612

Допустим, что вы нашли процедуру конца программы, которая вызывается после того, как исчерпано количество ошибок. Однако это середина процедуры, а надо найти ее начало. В поисках места, где появляется адрес BB90, просмотрите предшествующие байты.

C38F LD A, (B933)
DEC A
CP FF
JP Z BB90
LD (B933),



Знакомая последовательность, правда? Действительно, вы нашли место, в котором содержимое счетчика уменьшается на 1, после чего наступает проверка, достигло ли оно значения минус 1. Если да, то программа переходит к процедуре, выводящей на экран заключительные надписи.

Метод поисков с конца нередко более эффективен, чем чтение программы со стартового адреса, при котором приходится проверять, куда передается управление программой. Но он не дает результата, если вы не в состоянии найти процедуру, информирующую о выигрыше или поражении (последнее случается гораздо чаще).

Остается еще один способ - воспользоваться процедурами, связанными со звуком. В большинстве игр всяческим неприятным столкновениям сопутствует какой-то звук. Иногда это взрыв или скрежет, иногда - насмешливая мелодия. Ваша задача - найти места, из которых эти процедуры вызываются, и проанализировать соседние ячейки. Метод не очень эффективен и применяется только в крайних случаях, но если и он не даст результатов, то вас ожидает кропотливое и занимающее массу времени чтение всей программы.

А вот еще один пример того, как можно облегчить себе жизнь - на примере игры NODES OF YESOD.

Сначала "бессмертие" и отключение счетчика времени. В программе применено типичное решение:

86CC LD A, 03 только 3 "жизни"
LD (5B09), A

Интересно то, что программа показывает только одну цифру, даже если в счетчике ошибок 03 заменить, например, трехзначным FF (255).

7F92 LD HL, 5B09
7F95 DEC (HL)

и переход к концу, если A=0

7F5F LD A, (5B09)
DEC A
JP Z 9907

Обратите внимание, что здесь DEC служит исключительно для проверки. Результат не пересылается в 5B09. Счетчик времени на экране показывает часы, минуты и секунды. Время в программе идет быстрее, чем в действительности, и после достижения десяти часов игра кончается. В связи с этим на индикаторе счетчика времени час обозначается всегда одной цифрой, к которой стоит более внимательно присмотреться. Увеличивается она следующим образом:

9AB2 LD A, (5B0C)
INC A
DAA десятичная арифметика
LD (5B0C), A

и проверка предельного значения

7DC3 LD A, (5B0C)
CP 10 сравнение
JR C 7DCF С означает здесь "кредит", т.е. ситуацию, в которой от меньшего числа отнимается большее
7DCA LD A, 01 A=1 означает для процедуры конца превышение предела времени
JP 9907
7DCF NOP

Если A<10, то нужен "кредит", т.е. переход к адресу 7DCF - и игра продолжается. В противном случае наступает конец с надписью "OUT OF TIME". Можно это изменить, записав под адресом 7DCA новые команды:

7DCA LD A, 00
LD (5B0C), A

Таким образом, счетчик времени после достижения 10 начнет отсчет с 0.

Занесите 0 по адресу 7F95 (NOP вместо DEC (HL)), ликвидируйте ограничение времени и запустите измененную игру. Но вот - неприятный сюрприз. Приходит красный космонавт и забирает с таким трудом найденные предметы. Прежде чем вы вернетесь к тексту игры и попытаете его заблокировать, обратите внимание на то, что он появляется и тогда, когда у вас еще ничего нет. В такой ситуации при столкновении с вашим персонажем он исчезает, отнимая часть силы. Надо поискать место, где программа проверяет ваше "имущество". Вы находите следующее:

A77E LD A, (5B01)

A781 OR A

логическая сумма A и A,

результат равен 0 тогда и только тогда, когда A=0

переход, если 0

JR Z A7CD

A784 SET 0, (HL)

Можно заменить безусловный переход условным и тогда космонавт всегда будет уходить с пустыми руками. Исправляйте игру и запускайте ее снова. Цель - собрать одинаковые предметы и отнести их в указанное место. Допустим, вы собираете зеленые квадраты, но случайно взяли красное кольцо и не знаете, как от него избавиться. Изменить тип собираемых предметов или начать игру сначала? Тут вспоминается космонавт и вы начинаете жалеть, что вывели его из игры. Ведь он мог бы без особого вреда забрать это ненужное кольцо. Но теперь вы умеете изменить программу, и космонавт, прежде чем взять что-либо, попросит разрешения. Итак, найдите в программе свободное место (например, информационный текст, не имеющих значения для игры) и впишите собственную процедуру. Сначала ее вызов:

A77E JP 8CCA

с 8CCA начинается текст

и далее:

8CCA LD A, (5B01)

это надо повторить для ситуации, когда у вас ничего не будет

OR A

JP Z A7CD

8CD1 LD A, FB

проверка, была ли нажата клавиша [T]

IN A, (FE)

AND 10

JR NZ 8CDF

если нет, то проверка клавиши [N]

LD A, (5B01)

A получит соответствующее значение

JP A781

8CDF LD A, 7F

проверка клавиши [N]

IN A, (FE)

AND 08

JR NZ 8CD1

возврат к [T], так как не были нажаты ни [N], ни [T]

LD A, (5B01)

нажата [N]

JP A7CD

Теперь можете вернуться к игре и без опасений ожидать появления космонавта. После каждой встречи с ним программа будет останавливаться и ждать нажатия клавиши [T] (можно взять последний предмет) или N (ничего не брать). (Это "польский" вариант: T от польского слова ТАК, означающего "ДА", N - от слова NIE - "НЕТ". Если принцип понятен, можно ввести русские буквы - Примеч. пер.)

Если вам удалось успешно преобразовать представленные программы, то будьте готовы к новым неожиданностям. Программисты постоянно совершенствуют методы отравления нам жизни ограничениями времени, энергии или ошибок. Если профессия "взломщика" пришла к вам по душе, то можно уверенно сказать, что со временем вы сумеете "укротить" самые новые и самые трудные программы. Ну, а если вы поделитесь своим опытом с другими читателями, то все мы будем вам благодарны.

Мнемоники

микропроцессора Z-80

DEC SPECTRUM ASC Z80 CBH EDH

0	не использ.	nop	rlc B
1	не использ.	ld BC,nn	rlc C
2	не использ.	ld (BC),a	rlc D
3	не использ.	inc BC	rlc E
4	TRUE VIDEO	inc B	rlc H
5	INVERT VIDEO	dec B	rlc L
6	TAB,CAPS LOCK	ld B,n	rlc (HL)
7	EDIT	rlcA	rlc A
8	CRSR LT	ex AF,AF	rrc B
9	CRSR RT	add HL,BC	rrc C
10	CRSR DN	ld A,(BC)	rrc D
11	CRSR UP	dec BD	rrc E
12	DELETE	inc C	rrc H
13	ENTER	dec C	rrc L
14	NUMBER	ld C,n	rrc (HL)
15	GRAPH	rrcA	rrc A
16	INK	djnz DIS	rl B
17	PAPER	ld DE,(nn)	l C
18	FLASH	ld (DE),A	rl D
19	BRIGHT	inc DE	rl E
20	INVERSE	inc D	rl H
21	OVER	dec D	rl L
22	AT	ld D,n	rl (HL)
23	TAB	rlA	rl A
24	не использ.	jr DIS	rr B
25	не использ.	add HL,DE	rr C
26	не использ.	ld A,(DE)	rr D
27	не использ.	dec DE	rr E
28	не использ.	inc E	rr H
29	не использ.	dec E	rr L
30	не использ.	ld E,n	rr (HL)
31	не использ.	rrA	rr A
32	ПРОБЕЛ	jr nz,DIS	sla B
33	!	ld HL,nn	sla C
34	~	ld (nn),HL	sla D
35	#	inc HL	sla E
36	\$	inc h	sla H
37	%	dec H	sla L
38	&	ld H,n	sla (HL)
39	.	daa	sla A
40	(	jr z,DIS	sra B
41	)	add HL,HL	sra C
42	*	ld HL,(nn)	sra D
43	+	dec HL	sra E
44	,	inc L	sra H
45	-	dec L	sra L
46	:	ld L,n	sra (HL)
47	/	cpl	sra A
48	0	jr nc,DIS	
49	1	ld SP,nn	
50	2	ld (nn),A	
51	3	inc SP	
52	4	inc (HL)	
53	5	dec (HL)	
54	6	ld (HL),n	
55	7	scf	
56	8	jr c,DIS	srl B

57	9	add HL,SP	srl C
58	:	ld A,(nn)	srl D
59	:	dec SP	srl E
60	<	inc A	srl H
61	=	dec A	srl L
62	>	ld A,n	srl (HL)
63	?	ccf	srl A
64	@	ld B,B	bit 0,B
65	A	ld B,C	bit 0,C
66	B	ld B,D	bit 0,D
67	C	ld B,E	bit 0,E
68	D	ld B,H	bit 0,H
69	E	ld B,L	bit 0,L
70	F	ld B,(HL)	bit 0,(HL)
71	G	ld B,A	bit 0,A
72	H	ld C,B	bit 1,B
73	I	ld C,C	bit 1,C
74	J	ld C,D	bit 1,D
75	K	ld C,E	bit 1,E
76	L	ld C,H	bit 1,H
77	M	ld C,L	bit 1,L
78	N	ld C,(HL)	bit 1,(HL)
79	O	ld C,A	bit 1,A
80	P	ld D,B	bit 2,B
81	Q	ld D,C	bit 2,C
82	R	ld D,D	bit 2,D
83	S	ld D,E	bit 2,E
84	T	ld D,H	bit 2,H
85	U	ld D,L	bit 2,L
86	V	ld D,(HL)	bit 2,(HL)
87	W	ld D,A	bit 2,A
88	X	ld E,B	bit 3,B
89	Y	ld E,C	bit 3,C
90	Z	ld E,D	bit 3,D
91	[	ld E,E	bit 3,E
92	/	ld E,H	bit 3,H
93	]	ld E,L	bit 3,L
94	†	ld E,(HL)	bit 3,(HL)
95	-	ld E,A	bit 3,A
96	£	ld H,B	bit 4,B
97	a	ld H,C	bit 4,C
98	b	ld H,D	bit 4,D
99	c	ld H,E	bit 4,E
100	d	ld H,H	bit 4,H
101	e	ld H,L	bit 4,L
102	f	ld H,(HL)	bit 4,(HL)
103	g	ld H,A	bit 4,A
104	h	ld L,B	bit 5,B
105	i	ld L,C	bit 5,C
106	j	ld L,D	bit 5,D
107	k	ld L,E	bit 5,E
108	l	ld L,H	bit 5,H
109	m	ld L,L	bit 5,L
110	n	ld L,(HL)	bit 5,(HL)
111	o	ld L,A	bit 5,A
112	p	ld (HL),B	bit 6,B
113	q	ld (HL),C	bit 6,C
114	r	ld (HL),D	bit 6,D
115	s	ld (HL),E	bit 6,E

116	t	ld (HL),Hbit 6,H		186	INT	cp D res 7,D	indr
117	u	ld (HL),L bit 6,L		187	SQR	cp E res 7,E	otdr
118	v	halt bit 6,(HL)		188	SGN	cp H res 7,H	
119	w	ld (HL),Abit 6,A		189	ABS	cp L res 7,L	
120	x	ld A,B bit 7,B	in A,(C)	190	PEEK	cp (HL) res 7,(HL)	
121	y	ld A,C bit 7,C	out (C),A	191	IN	cp A res 7,A	
122	z	ld A,D bit 7,D	adc HL,SP	192	USR	ret nz set 0,B	
123	[	ld A,E bit 7,E	ld SP,(nn)	193	STR\$	pop BC set 0,C	
124	!	ld A,H bit 7,H		194	CHR\$	jp nz,nn set 0,D	
125	]	ld A,L bit 7,L		195	NOT	jp nn set 0,E	
126	-	ld A,(HL)bit 7,(HL)		196	BIN	call nz,nnset 0,H	
127	@	ld A,A bit 7,A		197	OR	push BC set 0,L	
128		add A,B res 0,B		198	AND	add A,n set 0,(HL)	
129		add A,C res 0,C		199	<=	rst 0 set 0,A	
130		add A,D res 0,D		200	>=	ret z set 1,B	
131		add A,E res 0,E		201	<	ret set 1,C	
132		add A,H res 0,H		202	LINE	jp z,nn set 1,D	
133		add A,L res 0,L		203	THEN	set 1,E	
134		add A,(HL) res 0,(HL)		204	TO	call z,nn set 1,H	
135		add A,A res 0,A		205	STEP	call nn set 1,L	
136		adc A,B res 1,B		206	DEF FN	adc A,n set 1,(HL)	
137		adc A,C res 1,C		207	CAT	rst 8 set 1,A	
138		adc A,D res 1,D		208	FORMAT	ret nc set 2,B	
139		adc A,E res 1,E		209	MOVE	pop DE set 2,C	
140		adc A,H res 1,H		210	ERASE	jp nc,nn set 2,D	
141		adc A,L res 1,L		211	OPEN #	out (n),A set 2,E	
142		adc A,(HL) res 1,(HL)		212	CLOSE #	call nc,nnset 2,H	
143		adc A,A res 1,A		213	MERGE	push DE set 2,L	
144	A (UDG)	sub B res 2,B		214	VERIFY	sub n set 2,(HL)	
145	B (UDG)	sub C res 2,C		215	BEEP	rst 16 set 2,A	
146	C (UDG)	sub D res 2,D		216	CIRCLE	ret c set 3,B	
147	D (UDG)	sub E res 2,E		217	INK	exx set 3,C	
148	E (UDG)	sub H res 2,H		218	PAPER	jp c,nn set 3,D	
149	F (UDG)	sub L res 2,L		219	FLASH	in A,(n) set 3,E	
150	G (UDG)	sub (HL) res 2,(HL)		220	BRIGHT	call c,nnset 3,H	
151	H (UDG)	sub A res 2,A		221	INVERSE	IXset 3,L	
152	I (UDG)	sbc A,B res 3,B		222	OVER	sbc A,n set 3,(HL)	
153	J (UDG)	sbc A,C res 3,C		223	OUT	rst 24 set 3,A	
154	K (UDG)	sbc A,D res 3,D		224	LPRINT	ret po set 4,B	
155	L (UDG)	sbc A,E res 3,E		225	LLIST	pop HL set 4,C	
156	M (UDG)	sbc A,H res 3,H		226	STOP	jp po,nn set 4,D	
157	N (UDG)	sbc A,L res 3,L		227	READ	ex (SP),HL set 4,E	
158	O (UDG)	sbc A,(HL) res 3,(HL)		228	DATA	call po,nn set 4,H	
159	P (UDG)	sbc A,A res 3,A		229	RESTORE	push HL set 4,L	
160	Q (UDG)	and B res 4,B	ldi	230	NEW	and n set 4,(HL)	
161	R (UDG)	and C res 4,C	cpi	231	BORDER	rst 32 set 4,A	
162	S (UDG)	and D res 4,D	ini	232	CONTINUE	ret pe set 5,B	
163	T (UDG)	and E res 4,E	outi	233	DIM	jp (HL) set 5,C	
164	U (UDG)	and H res 4,H		234	REM	jp pe,nn set 5,D	
165	RND	and L res 4,L		235	FOR	ex DE,HLset 5,E	
166	INKEY\$	and (HL) res 4,(HL)		236	GO TO	call pe,nn set 5,H	
167	PI	and A res 4,A		237	GO SUB	set 5,L	
168	FN	xor B res 5,B	ldd	238	INPUT	xor n set 5,(HL)	
169	POINT	xor C res 5,C	cpd	239	LOAD	rst 40 set 5,A	
170	SCREEN\$	xor D res 5,D	ind	240	LIST	ret p set 6,B	
171	ATTR	xor E res 5,E	outd	241	LET	pop AF set 6,C	
172	AT	xor H res 5,H		242	PAUSE	jp p,nn set 6,D	
173	TAB	xor L res 5,L		243	NEXT	diset 6,E	
174	VAL\$	xor (HL) res 5,(HL)		244	POKE	call p,nnset 6,H	
175	CODE	xor A res 5,A		245	PRINT	push AF set 6,L	
176	VAL	or B res 6,B	ldir	246	PLOT	or n set 6,(L)	
177	LEN	or C res 6,C	cdir	247	RUN	rst 48 set 6,A	
178	SIN	or D res 6,D	inir	248	SAVE	ret m set 7,B	
179	COS	or E res 6,E	otir	249	RANDOMIZE	ld SP,HL set 7,C	
180	TAN	or H res 6,H		250	IF	jp m,nn set 7,D	
181	ASN	or L res 6,L		251	CLS	eiset 7,E	
182	ACS	or (HL) res 6,(HL)		252	DRAW	call m,nnset 7,H	
183	ATN	or A res 6,A		253	CLEAR	IXset 7,L	
184	LN	cp B res 7,B	lddr	254	RETURN	cp n set 7,(HL)	
185	EXP	cp C res 7,C	cpdr	255	COPY	rst 56 set 7,A	

 компьютер дома 

© Тадеуш Радюш

Маленькие секреты TR-DOS версии 5.03

Дисковая операционная система — это способ хранения программ и файлов с данными в домашних и профессиональных компьютерных системах. По сравнению с традиционными "ленточными" системами, такими, как магнитофон или микродрайв, дисковые системы имеют множество преимуществ. Они более надежны, быстры и ими гораздо легче пользоваться.

В последнее время среди поклонников компьютера ZX Spectrum стала популярной дисковая операционная система TR-DOS (версии 5.03) фирмы Technology Research Ltd. Эта система коренным образом меняет возможности старого доброго компьютера Spectrum, превращая его из игрушки с довольно медленным и нудным процессом загрузки программ в весьма эффективный инструмент.

Самая краткая характеристика TR-DOS сводится к следующему:

ЛЕГКИЙ И БЫСТРЫЙ ДОСТУП — загрузка программ или данных происходит гораздо быстрее, чем с помощью магнитофона;

ШИРОКИЙ ВЫБОР ДИСКОВОДОВ — 5,25", 3,5" или 3", 40 или 80 дорожек, одно- или двусторонние дисководы. Другими словами, вы можете использовать большинство имеющихся на рынке дисководов;

ЕМКОСТЬ — до 4 дисководов одновременно на линии. Данные записываются с двойной плотностью, что обеспечивает максимальную емкость: до 2,5 млн. знаков;

ГИБКОСТЬ — интерфейс независим, так что вы можете использовать дисководы с другими компьютерами;

СОВМЕСТИМОСТЬ — интерфейс можно применять с компьютерами Spectrum, Spectrum+ и Spectrum 128;

АВТОСТАРТ — запускает Бейсик-программу интерфейса автоматически после включения питания или сброса (только в Spectrum и Spectrum+);

АВТОМАТИЧЕСКАЯ ПРОВЕРКА — автоматически проверяет и согласовывает систему со спецификацией дисковода;

"ВОЛШЕБНАЯ" КЛАВИША (MAGIC KEY) — благодаря ей вы можете мгновенно переписывать программы, загруженные в компьютер с магнитной ленты (или дисковода), на дискету;

СИСТЕМНЫЙ ПЕРЕКЛЮЧАТЕЛЬ — дает возможность отключать интерфейс или используется для сброса;

TR-DOS содержится в ПЗУ и использует только 12 байтов ОЗУ;

СИНТАКСИС — самый простой, с применением ключевых слов компьютера Spectrum, что позволяет осуществлять доступ к файлам на диске в TR-DOS из Бейсика или машинных кодов;

РАБОТА С ФАЙЛАМИ — TR-DOS работает с числовыми и знаковыми массивами, серийными (последовательными) файлами и файлами с истинно произвольным доступом.

Диски и дисководы

Интерфейс может использовать 5,25", 3,5" и 3" дисководы. Вы, несомненно, слышали такие названия, как флоппи-диск, дискета, мини-диск или микродиск. Мы будем называть их просто дисками.

В настоящее время наиболее популярны 5,25" диски. Все же имеется тенденция к более маленьким 3" дискам, хотя они дороже и менее надежны.

TR-DOS позволяет разбить дорожку на 16 секторов по 256 байт в каждом. Такое большое количество секторов малой величины имеет определенное преимущество. Во-первых, если нужно сохранить лишь некоторые данные, то будет использовано незначительное пространство диска, что в конечном итоге позволит записать большее число файлов. Во-вторых, при использовании файлов произвольного доступа появляется возможность сделать программу более гибкой и увеличить скорость операций.

Нужно отметить, что дорожка 0 (внешняя) в TR-DOS предназначена для хранения системной дисковой информации.

Сказанное выше дает нам возможность подсчитать количество секторов и вместимость форматированного диска (SS — односторонний, DS — двусторонний):

40 дорожек SS = $38 * 16 = 624$ сектора * 256 = 156 Кбайт
 40 дорожек DS = $79 * 16 = 1264$ сектора * 256 = 316 Кбайт
 80 дорожек SS = $79 * 16 = 1264$ сектора * 256 = 316 Кбайт
 80 дорожек DS = $159 * 16 = 2544$ сектора * 256 = 636 Кбайт

Это означает, что на 1 дорожке помещается 4 Кбайта или 4 сектора на 1 Кбайт. Таким образом, разделив количество свободных секторов на 4, можно узнать, сколько килобайт свободного пространства осталось на диске.

Способ работы со стандартными командами TR-DOS приведен в прилагаемой к ней фирменной документации. При программировании на Бейсике пользоваться ими легко и удобно. Однако, выполняя более сложную работу на компьютере Spectrum, приходится обращаться к операционной системе, минуя Бейсик и интерпретатор TR-DOS, например, из программы, написанной в машинных кодах или на другом языке. Предлагаемый в фирменном руководстве к TR-DOS способ (имитация строки Бейсика в кодах) громоздок и неудобен. Гораздо лучше обращаться непосредственно к операционной системе, находящейся в "подставном" ПЗУ. Для этого в TR-DOS существуют специальные возможности, и на некоторых из них мы остановимся.

Выходные точки TR-DOS

Для обращения к операционной системе TR-DOS существует специальная точка входа с адресом 15663: за командой NOP следует единственная команда RET (код 201). Это позволяет осуществлять переходы по любому требуемому адресу, предварительно помещенному на вершину стека. Этот способ хорош в том случае, когда есть дисассемблированный текст программы TR-DOS с отмеченными адресами соответствующих подпрограмм. Любопытным и пытливым читателям можно рекомендовать в качестве упражнения получить такой текст и разобраться в нем самостоятельно. Необходимые для этого исходные коды можно записать (как читатель уже, вероятно, догадался) в виде файла типа CODE: это первые 16384 байта памяти, записанные на диск из операционной системы: SAVE "TR-DOS" CODE 0,16383. Однако следует предупредить, что интерпретатор DOS написан крайне сложно и невразумительно. Чтобы облегчить расшифровку кодов, отметим участки, которые занимают тексты и таблицы, а также некоторые подпрограммы (адреса даны в десятичном виде):





000102 - 000105	переход на адрес 10838 (обслуживание клавиши Magic)
000864 - 000939	название версии TR-DOS
002048 - 004095	свободные ячейки памяти
004099 - 004119	текст "Подсоединен интерфейс 1"
004261 - 004432	текст системной информации для LIST
004445 - 004519	подпрограмма печати целого листа (регистр XL) в десятичной форме
008121 - 008137	таблица
010086 - 010299	текст сообщений TR-DOS
010380 - 010455	таблица адресов подпрограмм, адресуемых регистром C
010673 - 010804	текст сообщений TR-DOS
010838 - 012038	обслуживание клавиши Magic
012275 - 012337	таблица
012541 - 012786	текст - ключевые слова TR-DOS
012797 - 015360	свободные ячейки
015610 - 015664	точки входа в TR-DOS

При непосредственном обращении к TR-DOS наиболее важной является точка входа с адресом 15635 (#3D13). С этого адреса в конечном итоге мы переходим к подпрограмме 10300, которая в зависимости от кода, содержащегося в регистре C, при помощи таблицы 10380-10455 передает управление соответствующей подпрограмме-процедуре. Параметры для этих подпрограмм сообщаются при помощи регистров микропроцессора либо находятся в области системных переменных TR-DOS (например, заголовков файла на диске). Ниже приводятся краткие описания процедур, вызываемых в зависимости от содержимого регистра C (заглавными буквами обозначено содержимое соответствующих регистров):

- 1) C=05 - осуществляется чтение с диска: в буфер по адресу XL считывается B секторов, первый из которых имеет номер E на дорожке D; регистр A=0;
- 2) C=06 - из буфера по адресу XL на диск (дорожка A, сектор E) записывается B секторов; A=255;
- 3) C=07 - чтение каталога с диска; A содержит номер каталога, по которому высылается печать каталога;
- 4) C=08 - считывается заголовок с номером A в области системных переменных по адресу 23773-23788 (если не подключен интерфейс 1);
- 5) C=09 - то же, что и п. 4, но с записью заголовка на диск;
- 6) C=10 - поиск файла, имя которого находится в области системных переменных; номер найденного файла возвращается в регистр C;
- 7) C=11 - запись на диск файла типа CODE длиной DE с адреса XL; имя файла должно находиться в системной области памяти;
- 8) C=12 - запись программы в Бейсике; имя файла в системной области;
- 9) C=14 - загрузка программы в кодах; при этом заголовок также должен быть сформирован и помещен в ячейки 23773-23788, а в зависимости от содержимого регистра A подпрограмма будет работать по-разному.
- A=0 - коды загружаются с адреса, указанного в заголовке;
- A=3 - коды загружаются с адреса XL, причем длина загружаемого файла определяется значением DE;
- A=255 - коды загружаются с адреса XL, но длина загруженного файла равна длине файла, записанного на диск;
- 10) C=18 - стирается файл, порядковый номер которого на диске равен A.

Кроме того, при вызове подпрограммы с адресом 10469 (например, косвенно через стек и точку входа 15663) происходит перемещение заголовка файла (16 байтов) с адреса XL в адрес 23773; при этом регистр A должен содержать 0.

Структура заголовка файла такова: 8 байтов занимает имя; 1 байт - тип (B,C,D или *); 2 байта - начальный адрес; 2 байта - длина файла; 1 байт - занимаемое количество секторов; 2 последних байта - соответственно номер сектора и дорожки, с которых начинается файл на диске.

На диске заголовки файлов записаны в первых 8 секторах (с 0 по 7) нулевой дорожки. Системную информацию о диске содержит 8-й сектор нулевой дорожки, точнее, последние байты этого сектора, начиная с номера 225. Ниже приводятся адреса и их содержимое в системном секторе:

225 -	первый свободный сектор
226 -	первая свободная дорожка
227 - 22 -	двусторонний, 24 - односторонний диск
228 -	количество файлов
229, 230 -	количество свободных секторов (младший и старший байты)
231 -	количество секторов в дорожке (16)
232 -	0
233 -	0
234 -	:
...	> коды пробела (32 в ASCII)
242 -	:
243 -	0
244 -	количество удаленных файлов
245 -	:
...	> имя диска из 8 символов
252 -	:
253 -	0
254 -	0
255 -	0

Кроме того, следует заметить, что порядковый номер заголовка на нулевой дорожке имеет следующее свойство: старший байт соответствует номеру сектора, в котором записан заголовок, а младший - номеру заголовка в секторе. Это иногда полезно учитывать при работе с диском.

В заключение отмечу, что сейчас, благодаря использованию изложенных в статье принципов, появляется все больше программ для Spectrum, адаптированных к работе с TR-DOS, например язык Паскаль версии HP4TM16 (адаптированный вариант называется Pascal D), редактор текстов Tasword-2 и некоторые другие.

Московская Школьная Телекоммуникационная Сеть (MoSTNet) - первая в Советском Союзе система компьютерной коммуникации для обслуживания образовательных учреждений. Она начала действовать осенью 1989 г. и первоначально поддерживала 12 московских школ, участвующих в NYS/Moscow STP. Сегодня абонентами системы являются школы Москвы, Киева и Ленинграда, работники Московского городского комитета по народному образованию и МГУ. Возможности расширения числа абонентов сети ограничены отсутствием в школах модемов и компьютеров.

На хост-компьютере MoSTNet используется пакет WildCat. Для выхода в международные компьютерные сети MoSTNet использует коммутационный узел IAS в Москве.

Основная задача MoSTNet - отработка эффективных способов использования средств компьютерной коммуникации в образовании. На базе MoSTNet работает Лаборатория "Телекоммуникация в образовании" Научного Совета АН СССР по комплексной проблеме "Кибернетика", ведется разработка учебных материалов.

Руководитель MoSTNet - Александр Юрьевич Уваров.

Для связи с MoSTNet можно использовать следующие адреса:

SFMT:MoSTNet, EIES:1593, IASNet:MoSTNet
Доступ через BITNET, INTERNET

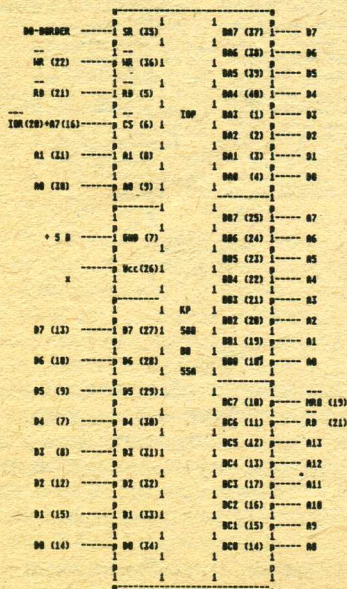
© Игорь Щетинин

Эмулятор ПЗУ для ZX Spectrum

При разработке микропроцессорных устройств (компьютеров, средств автоматизации эксперимента и т.п.) нарисовать электрическую схему всегда проще, чем написать затем для этого устройства управляющую программу. Такая программа хранится в постоянном запоминающем устройстве (ПЗУ). От того, как она написана, будет зависеть "дружелюбность" вашего детища. Поскольку в процессе отладки обнаруживаются ошибки, да и в голову лезут еще более гениальные мысли, информацию в ПЗУ приходится корректировать. Перепрограммирование ПЗУ с ультрафиолетовым стиранием — процесс достаточно нудный, а сообщения о появлении отечественных ПЗУ с электрическим стиранием — скорее слухи (как, впрочем, и сообщения о микропроцессоре 1810BM80). В этих случаях используют эмулятор ПЗУ.

Эмулятор ПЗУ — это обычно компьютер, который обрабатывает адресные сигналы и сигналы разрешения считывания и выдает информацию на шину данных устройства в зависимости от их комбинации. Правда, такое "ПЗУ" обладает крайне большой "задержкой", что делает его непригодным для отладки устройств реального времени (например, цифровая звукозапись). В подобных случаях удобно в качестве "ПЗУ" использовать оперативное запоминающее устройство (ОЗУ) и программировать его с помощью компьютера.

В случае применения ОЗУ, в качестве компьютера совсем необязательно использовать относительно мощные IBM или ДВК — вполне подойдет и Спектрум. Его лишь необходимо дополнить (если это еще почему-то не сделано) параллельным портом 580BB55A (в дальнейшем — просто "порт"), подключенным, как показано на рисунке.



01-BORDER 0058 (25)
02-BORDER 0059 (26)

К разрабатываемому устройству

К Спектруму

Внимание: сигналы IOR и A7 на вход CS порта подаются через вентиль 2-ИЛИ, входящий в состав микросхемы 555ЛЛ1. Помимо выводов порта используются также выходы внутреннего порта 254 (он управляет цветом рамки — в дальнейшем "BORDER"). Для простоты полагается, что ваше устройство также строится на Z80, в противном случае сигналы подаются на аналогичные входы применяемого микропроцессора (в скобках указаны номера выводов для Z80 и 580BB55A). Кроме того, подразумевается, что микросхема ОЗУ установлена вместо реального ПЗУ на уже спаянном макете устройства. Например, вместо ПЗУ 573PФ2 удобно подставлять ОЗУ 537PУ10, разводка которых совпадает.

Работа пойдет гораздо быстрее, если помимо порта вы сможете найти Ассемблер для выбранного микропроцессора (например, ZEUS фирмы Crystal Computing для Z80). Тогда можно с магнитофона или дисководов загружать тексты Ассемблера и листинг, дополнять и корректировать программу, транслировать ее и через порт записывать полученные коды в "ПЗУ". Хотя процессом записи можно управлять из Бейсика с помощью операторов IN, OUT и BORDER, лучше составить программу-загрузчик в кодах. Ниже приводится ее текст на Ассемблере для Z80:

```
10      ORG 64344
20      LD BC, число байтов для
          пересылки в "ПЗУ" + 1
30      LD HL, адрес начала массива
          кодов в памяти Спектрума
40      LD DE, 16388
50      LD A,E
60      OUT 254,A
70      LD A,128
80      OUT 3,A
90      ADD HL,BC
100     AA  DEC HL
110     DEC BC
120     LD A,(HL)
130     OUT 0,A
140     LD A,C
150     OUT 1,A
160     LD A,B
170     ADD A,D
180     OUT 2,A
190     XOR A
200     OUT 254,A
210     LD A,E
220     OUT 254,A
230     XOR A
240     XOR B
250     JR NZ, AA
260     XOR C
270     JR NZ, AA
280     RET
```

Загрузчик запускается командой PRINT USR 64344. О выигрыше во времени говорит тот факт, что 35-байтовый фрагмент программы через Бейсик загружался около 2 с, тогда как через вышеописанный загрузчик программа объемом 2048 байт загружается почти в 4 раза быстрее.

Если же Ассемблера нет, то следует сначала вручную оттранслировать программу, а затем оформить ее в Бейсике как данные

4 DATA 243, ...

При этом коды для каждой команды (например, LD B, 8) рекомендуются размещать в отдельной строке, строки нумеровать с шагом 10 и следить, чтобы номера не пересекались. Тогда, написав следующий загрузчик:

1 MERGE "": GO TO 1

2 RESTORE K: BORDER 4:OUT 3,128:FOR N=0 TO 63:FOR M=0 TO 255:READ K:OUT 0,K:OUT 1,M:OUT 2,(N+64):BORDER 0: BORDER 4:NEXT M:NEXT N



и запустив выполнение строки 1, можно "слить" фрагменты программы в единое целое. После этого командой BREAK прервать выполнение строки 1 и запустить программу со строки 2. Программа запишет данные в "ПЗУ" и завершит выполнение сообщением "E OUT OF DATA". Кстати, таким же образом вы можете оттранслировать и загрузчик.

При желании можно проверить правильность записи информации. Это будет особенно полезно на первых порах, когда будут не только "всплывать" ваши собственные ошибки в монтаже, но и выявляться плохие контакты между шлейфом и разъемами. Подставляя другие адреса, можно читать информацию из ОЗУ:

```
3 BORDER 4: OUT 3,144: FOR N=0 TO 63:FOR M=0 TO 255:OUT
2,N:OUT 1,M:PRINT IN 0:NEXT M:NEXT N
```

Всего этого достаточно, чтобы при наличии осциллографа даже в домашних условиях разработать устройство, но, если вы захотите, можно пойти и дальше. Например, поставить второй порт и увеличить количество двуполупроводников шин. Можно также использовать выход D4 порта 254 в качестве генератора для потактовой отладки устройства с контролем состояния шин через порт. Управляющие программы пишутся по аналогии с вышеприведенными. Обратите внимание на то, что переключить порт на работу в режиме "МОНИТОР" (все каналы работают на считывание) можно не только с помощью записи управляющего слова в порт 3, но и командой BORDER 7 (при этом на экране телевизора виден режим работы порта).

Желающим познакомиться с эмулятором ПЗУ подробнее рекомендуется книга: Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник. В.-Б.Б. Абрайтис и др. - М.: Радио и связь, 1988. - Т.1. - с.82-90.

Конструкция

Генератор звука AY-3-8910 кроме своего основного назначения может исполнять роль устройства ввода/вывода, так как он содержит два дополнительных восьмиразрядных порта ввода/вывод. Считывание и запись данных из этих портов не влияют на процессы формирования звука. Порты должны быть целиком (по 8 бит) определены либо для считывания, либо для записи. Используя эти порты, можно построить параллельный интерфейс, предназначенный для подключения принтера. Через один порт, настроенный на вывод, в принтер будут пересылаться коды печатаемых символов, а через другой порт попеременно будет то выдаваться сигнал STROBE, управляющий приемом информации принтером, то приниматься сигнал занятости принтера BUSY. Временные диаграммы этих сигналов представлены на рис. 1.

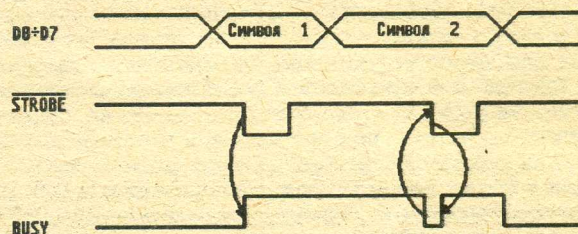


Рис. 1

Прием принтером стробирующего импульса вызывает выдачу им сигнала занятости, а также считывание состояния шины данных. Очередной стробирующий сигнал может появиться только после исчезновения сигнала занятости, и может длиться какое угодно время. Например, пользователь может привести принтер в это состояние нажатием кнопки BUSY, скажем, для замены листа бумаги. Выключение сигнала занятости ведет к возобновлению передачи данных (а значит, и печати) без потери переданных символов.

Принципиальная схема всего интерфейса показана на рис. 2. Выводы схемы AY-3-8910 под номерами 3, 4, 22 и 38 относятся только к его звуковым функциям, и их подключение здесь описано не будет.

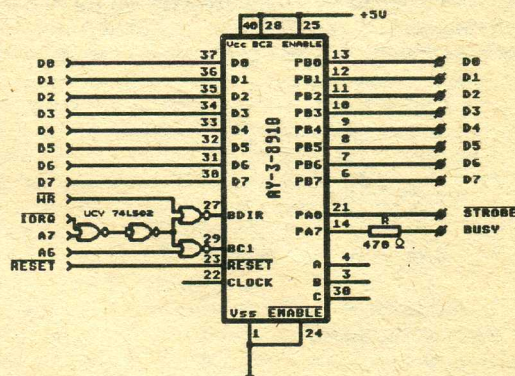


Рис. 2

Выводы 27 и 29 управляют обменом данными с микропроцессором и видом исполняемой операции: BDIR=0, BC1=0 - схема не связана с микропроцессором; BDIR=0, BC1=1 - считывание из ранее указанного регистра; BDIR=1, BC1=0 - запись в ранее указанный регистр; BDIR=1, BC1=1 - запись адреса регистра. Во время установления связи микропроцессора с интерфейсом разряды A0-A5 адресной шины должны быть в состоянии "1", чтобы не мешать взаимодействию микропроцессора со схемой ULA и другими внутренними устройствами. С декодированием вводов BDIR и BC1 (рис. 2) связаны следующие команды доступа к интерфейсу:

Компьютер дома

© Гжегож Шмит

ZX Spectrum

и принтер

Практически все пользователи ZX Spectrum уже знают, что непосредственно к нему можно подключить только простой, одноигольный ZX PRINTER или специализированный принтер SEIKOSHA GP50. Для работы с другими принтерами, к сожалению, требуется соответствующий интерфейс. Это может быть фирменный INTERFACE 1 или самодельный простой параллельный интерфейс. Однако все они - 7-разрядные, из-за чего нельзя полностью использовать возможности подсоединенного принтера, в том числе и графический режим. По этой же причине в программное обеспечение таких интерфейсов не включена подпрограмма копирования экрана в режиме точечной печати.

В предлагаемой вашему вниманию статье излагается решение главной проблемы: описывается 8-разрядный параллельный интерфейс стандарта Centronics, построенный на основе программируемого генератора звука типа AY-3-8910, а также программа, обеспечивающая работу микрокомпьютера ZX Spectrum, снабженного этим интерфейсом, с популярными матричными принтерами фирмы STAR.

OUT(63),г - выбор регистра;
 OUT(95),п - запись данных в регистр;
 IN D,(63) - чтение данных из регистра.

Из 16 регистров схемы AY-3-8910 функции, связанные с интерфейсом, выполняют: r7 - управляющий регистр, r14 - регистр порта A, r15 - регистр порта B. Через порт B будут пересылаться 8-разрядные коды символов для печати, а через порт A - сигналы, управляющие передачей.

Управляющий регистр позволяет менять режим работы остальных регистров. Запись в него величины 191 устанавливает регистр PA на ввод, а регистр PB - на вывод, тогда как 255 переключает оба регистра, PA и PB, на вывод. Необходимо обратить особое внимание на линию BUSY. Со стороны принтера эта линия является выводной, тогда как со стороны интерфейса, в зависимости от режима работы порта PA, она может работать как на вывод, так и на ввод. Возможные конфликты на этой линии предотвращает регистр P, предохраняя от повреждения обе ее стороны.

Программное обеспечение

Программа обслуживания принтера была написана исходя из следующих требований:

- * работа с принтерами STAR;
- * реализация команд LLIST и LPRINT в непосредственном режиме;
- * копирование экрана в графическом режиме через RANDOMIZE USR;
- * табуляция печати разделителем "," (запятая);
- * редактирование печати: 64 знака в строке, 60 строк на странице;
- * возможность пересылать на принтер управляющие коды;
- * минимальный размер программы.

Из управляющих кодов 0-31 предусмотрено обслуживание только кода 6 (табуляция печати), а также кода 13 (ENTER). Вместо обслуживания остальных управляющих кодов ZX Spectrum (например, TAB, PRINT и т.д.), которые и так составляют небольшую часть управляющих кодов принтеров STAR, предусмотрен специальный режим пересылки управляющих кодов: получение принтером символа © с номером 127, отсутствующего в коде ASCII, расценивается как сообщение о том, что следующие два пересылаемых символа надо трактовать как число в пределах 0 - 225 в шестнадцатеричном виде. Таким образом с помощью последовательности ©хх©хх... можно переслать принтеру любой управляющий код.

Такое решение имеет ценное преимущество: оно позволяет пересылать управляющие коды, помещенные в предназначенный для печати текст, который подготовлен с помощью текстового редактора. Это делает возможными текущие изменения параметров печати. Естественно, можно пересылать управляющие коды непосредственно с клавиатуры командой LPRINT "©хх©хх...". Работа ZX Spectrum с принтером требует записи адреса программы обслуживания принтера в ячейки памяти 23749 и 23750, адресующие обслуживание третьего канала. На распечатке представлена программа обслуживания принтера, написанная на языке Ассемблера Z80.

В строке 10 необходимо поместить адрес начала программы. Программа может быть помещена в любом месте памяти, в зависимости от требований основной программы, с которой она должна работать. В строках 20-120 записывается процедура инициализации программы и интерфейса: строки 20 и 30 записывают адрес начала собственной программы в то место, где должен находиться адрес обслуживания третьего канала, строки 40-70 - начальные значения в три ячейки памяти, используемые программой, строки 80-120 настраивают порт A интерфейса на ввод, а порт B - на вывод. Программа применяет три ячейки памяти ОЗУ для хранения служебных данных. В ячейке с адресом N.LINE запоминается количество свободных строк в пределах страницы, в ячейке по адресу MEM - старший байт управляющего кода, а по адресу POS запоминается позиция печатаемого символа в пределах строки. Кроме того, при пересылке кодов управления принтером старшие биты ячеек N.LINE и MEM исполняют роль флаж-

ков. В противном случае происходит фильтрация кодов, посылаемых процессором (строки 570-680). Из диапазона 0-31 отбираются только коды 6 и 13. Печатные знаки (коды 32-126) передаются принтеру подпрограммой LINE, которая проверяет позицию печати в пределах строки. Получение кода 127 (символ ©) запоминается путем установления двух флажков в подпрограмме ESC. Ключевые слова языка Бейсик ZX Spectrum (так называемые "туокенз") обозначаются кодами 165-255. Из значения этих кодов вычитается число 165, в результате чего получается порядковый номер ключевого слова. После этого управление передается ПЗУ Spectrum, по адресу 0C10, где помещается подпрограмма идентификации ключевого слова на основе его порядкового номера и печати всего слова с требуемыми пробелами (предшествующими или последующими).

В строках 130-170 расположена подпрограмма обслуживания запятой, используемой в качестве знака табуляции печати. Она перемещает печатающую головку принтера в ближайшую позицию, номер которой делится на 16. Для этого принтеру посылается соответствующее число пробелов. В строках 180-220 устанавливаются два флажка, свидетельствующие о получении кода 127.

Строки 230-240 посылают принтеру код 27, который соответствует несуществующей в ZX Spectrum клавише ESCAPE.

В строках 250-400 помещается подпрограмма, которая заменяет 2 последовательно полученных знака (рассматриваемых как число в шестнадцатеричном виде) на десятичное число в пределах 0-255, пересылает это число на принтер и одновременно снимает установленные флажки.

В строках 410-520 размещается процедура обслуживания кода 13 (ENTER). На принтер посылаются по очереди коды 13 (CR - возврат каретки) и 10 (LF - перевод строки), обновляется позиция печати в строке, а также проверяется номер строки в пределах страницы. При достижении конца страницы еще 12 раз посылается код LF (для прогона бумаги к новой странице), а счетчик свободных строк устанавливается на значение 60.

В строках 760-890 находится основная подпрограмма обслуживания интерфейса, которая выдает содержимое аккумулятора на шину принтера. Выбрав регистр B (OUT(63),15), в него пересылается содержимое аккумулятора (OUT(95),A), а затем выбирается регистр A (OUT(63),14) для проверки состояния линии BUSY (IN D,(63)). Состояние "0" на линии BUSY дает возможность выдать сигнал стробирования данных. Этот импульс можно выдать упрощенным способом, без фактической пересылки данных в регистр A интерфейса. Дело в том, что проверка работы схемы AY-3-8910 показала, что выводы регистра, настроенного на чтение, ведут себя так, будто находятся в высоком состоянии (благодаря внутренним регистрам, подключенным к питанию), тогда как выходы регистра после установки его в режим вывода, переходят в низкое состояние. Поэтому подачу стробирующего импульса можно произвести, переключив на некоторое время порт A на режим вывода (строки 830 и 840), после чего строка 850 обеспечит возвращение интерфейса в первоначальное состояние. В следующей строке помещен вызов из ПЗУ микрокомпьютера ZX Spectrum подпрограммы проверки нажатия на клавишу BREAK. Если эта клавиша нажата, печать прерывается и на экран высылается соответствующее сообщение (строки 880-890).

Со строки 900 начинается подпрограмма копирования экрана в графическом режиме. После того как на принтер высылается последовательность "ESC 51 23" (которая устанавливает такой интервал между строками, что они печатаются вплотную), в регистровую пару HL загружается адрес начала области экранной памяти. Затем 24 раза (т.е. столько раз, сколько строк на экране), выполняется подпрограмма COPY. Далее на принтер высылается последовательность "ESC 64", восстанавливающая исходное состояние (инициализация принтера).



В строках 1130-1410 - подпрограмма точечной печати одной строки. Она начинается с посылки на принтер команды "графика нормальной плотности" (т.е. точечная печать в режиме одинарной плотности) в виде последовательности "ESC 75 0 1", затем 32 раза (количество символов в строке) выполняется подпрограмма COPSCN (строки с 1200 по 1360). После печати каждой строки посылаются коды 13 (CR) и 10 (LF) - строки 1380-1410. В пределах каждого знака 8 раз (количество колонок пикселей в знаке) выполняется подпрограмма CORVUT (строки 1210-1320). Области экрана, содержащей 1 знак, в экранной памяти соответствуют 8 байтам, по 1 на каждую колонку пикселей поля знака. Формирование колонок пикселей производится в строках 1210-1270.

В строках 1420-1440 происходит адресация трех ячеек памяти, используемых программой обслуживания принтера для хранения служебных данных.

В рамках всей подпрограммы копирования экрана регистровая пара HL содержит текущий адрес экранной памяти. Все операции, выполняемые на регистрах H и L, непосредственно следуют из организации экранной памяти и, пожалуй, не требуют подробных объяснений. Любопытный читатель легко может разобраться в этом, принимая во внимание организацию экранной памяти:

- * в битах b0-b4 регистра L закодирована позиция знака в строке (0-31);
- * номер строки пикселей в пределах одной строки записан в битах b0-b2 регистра H;
- * все изображение, состоящее из 24 строк, разделено на три сегмента по восемь строк;
- * сегменты адресованы битами b3-b4 регистра H;
- * номер строки в пределах сегмента закодирован битами b5-b6 регистра L.

Запуск программы

Загружаем в память компьютера ассемблер (например, GENS3M2), старательно переписываем приведенную программу и ассемблируем ее, предварительно указав адрес, который обозначен меткой COPSCR (строка 900). Для приведенного в качестве примера адреса начала программы (ORG 65000) метка COPSCR получит значение 65160. Подключаем принтер, переходим в Бейсик и инициализируем программу командой RANDOMIZE USR 65000. Теперь можем испытать работу команд LLIST и LPRINT (они аналогичны командам LIST и PRINT, но печать производится не на экране, а на принтере). Копию экрана получаем по команде RANDOMIZE USR 65160. К сожалению, это будет копия не всего экрана, а только его верхних 22 строк. Правда, можно записать короткую программу на Бейсике, которая считывает все изображение в память (LOAD SCREEN) и затем распечатывает его в точечном режиме. Полученную в результате ассемблирования программу можно записать на кассету, а затем присоединять к любой программе, которая предусматривает использование принтера.

Рассмотрим способ получения распечаток ассемблера GENS3M2 и редактора текста TASWORD. В GENS3M2 достаточно поместить программу обслуживания в произвольном месте памяти (необходимо следить только, чтобы она не пересекалась с GENS или областью кодов, получаемых в результате ассемблирования) и инициализировать ее, чтобы можно было пользоваться инструкцией "W" ассемблера и режимом ассемблирования под номером 8. В случае редактора TASWORD программу необходимо сократить (убрать ненужные в этом случае строки 900-1370), а также удалить проверку конца строки (строки 720-740 программы). Преобразованную таким образом программу ассемблируем с адресом начала 31800 и записываем как дополнительный блок программы TASWORD после первого блока на Бейсике. Сам первый блок также необходимо несколько изменить: в строке 15 команду CLEAR 31999 заменяем на CLEAR 31800, а в строке 17 дописываем второй раз LOAD "" CODE и после инструкции LET a=USR 59081: дописываем RANDOMISE USR 31800.

В заключение - замечание, предназначенное для владельцев других интерфейсов (построенных, например, с использованием микросхемы Intel 8255 либо ее советского аналога) или принтеров других типов. Программа написана таким образом, что достаточно заменить в ней фрагменты, касающиеся интерфейса или принтера. Возможность редактирования печати произведена с учетом использования других принтеров, поскольку в принтерах STAR эти функции обеспечивают соответствующие управляющие коды.

Программа обслуживания принтера STAR

10		ORG	65000
20		LD	HL,STAR
30		LD	(23749),HL
40		LD	HL,60
50		LD	(N.LINE),HL
60		XOR	A
70		LD	(POS),A
80	SETINT	LD	A,191
90	OUT1	LD	BC,#73F
100	OUT2	OUT	(C),B
110		OUT	(93),A
120		RET	
125			
130	COMMA	CALL	SPAC
140		LD	A,(HL)
150		AND	13
160		JR	NZ,COMMA
170		RET	
175			
180	ESC	DEC	HL
190		SET	7,(HL)
200		DEC	HL
210		SET	7,(HL)
220		RET	
225			
230	ESCAPE	LD	A,27
240		JR	PRINT
245			
250	PR_CNT	CP	#40
260		JR	C,HEX
270		SUB	7
280		HEX	AND #F
290		BIT	7,(HL)
300		JR	Z,HEX_2
310		RLA	
320		RLA	
330		RLA	
340		RLA	
350		LD	(HL),A
360		RET	
370	HEX_2	ADD	A,(HL)
380		DEC	HL
390		RES	7,(HL)
400		JR	PRINT
405			
410	NEWLIN	CALL	CRLF
420		LD	(HL),0
430		DEC	HL
440		DEC	HL
450		DEC	(HL)
460		RET	NZ
470		LD	(HL),60
480		LD	E,12
490	PAGE	CALL	LF
500		DEC	E
510		JR	NZ,PAGE
520		RET	
525			
530	STAR	LD	HL,N.LINE
540		BIT	7,(HL)
550		INC	HL
560		JR	NZ,PR_CNT
570		CP	6
580		INC	HL
590		JR	Z,COMMA
600		CP	13
610		JR	Z,NEWLIN
620		CP	32
630		RET	C
640		CP	127
650		JR	Z,ESC
660		JR	C,LINE

670		SUB	165
680		JP	NC, #C10
690	SPAC	LD A	, #20
700	LINE	PUSH	AF
710		INC	(HL)
720		LD	A, (HL)
730		CP	65
740		CALL	NC, NEWLIN
750		POP	AF
755			
760	PRINT	LD	BC, #F3F
770		CALL	OUT2
780		DEC	B
790		OUT	(C), B
800	BUSY	IN	D, (C)
810		INC	D
820		JR	Z, BUSY
830		LD	A, 255
840		CALL	OUT1
850		CALL	SETINT
860		CALL	#IF34
870		RET	C
880		RST	8
890		DEFB	20
895			
900	COPSCR	CALL	ESCAPE
910		LD	A, 51
920		CALL	PRINT
930		LD	A, 23
940		CALL	PRINT
945			
950		LD	B, 24
960		LD	HL, #4000
970	COPY	PUSH	HL
980		PUSH	BC
990		CALL	COPLIN
1000		POP	BC
1010		POP	HL
1020		LD	A, 1
1030		ADD	A, #20
1040		LD	L, A
1050		SBC	A, A
1060		AND	8
1070		ADD	A, H
1080		LD	H, A
1090		DJNZ	COPY
1100		CALL	ESCAPE
1110		LD	A, 64
1120		JR	PRINT
1125			
1130	COPLIN	CALL	ESCAPE
1140		LD	A, 75
1150		CALL	PRINT
1160		XOR	A
1170		CALL	PRINT
1180		LD	A, 1
1190		CALL	PRINT
1195			
1200	COPCHA	LD	E, 8
1210	COPBYT	RLC	(HL)
1220		RL	C
1230		INC	H
1240		LD	A, H
1250		AND	7
1260		JR	NZ, COPBYT
1270		LD	A, C
1280		CALL	PRINT
1290		LD	A, H
1300		SUB	8
1310		LD	H, A
1320		DEC	E
1330		JR	NZ, COPBYT
1340		INC	L
1350		LD	A, L
1360		AND	31
1370		JR	NZ, COPCHA
1375			
1380	CRLF	LD	A, 13
1390		CALL	PRINT
1400	LF	LD	A, 10
1410		JR	PRINT
1415			
1420	N.LINE	EQU	\$
1430	MEM	EQU	#+1
1440	POS	EQU	#+2

Centronics

для Atari XL/XE

Предлагаем простой способ подключения принтера в стандарте Centronics к домашним компьютерам Atari XE/XL. Предоставленную редакцией программу обслуживания печатающего устройства (решение 1) тестировал Януш Вишневский. В связи с некоторыми замечаниями он представил собственное предложение (решение 2).

РЕШЕНИЕ 1

Компьютеры Atari XE/XL в стандартном варианте не оборудованы разъемом Centronics, который позволяет работать с типовыми печатающими устройствами. Здесь представлен простой способ решения этой проблемы.

Для соединения компьютера с печатающим устройством необходимо использовать три разъема типа D (два 9-штырьковых и один 25-штырьковый).

Коммуникация происходит через порт А микросхемы PIA, который обычно используется для подсоединения джойстиков 1 и 2. Семь битов (A0-A6) этого порта служат для выхода данных: бит A7 - выход сигнала STB для печатающего устройства. Занятость принтера (сигнал BUSY) указывается состоянием входа TRIGO (клавиша джойстика 1).

Представленная управляющая программа позволяет принтеру работать в текстовом режиме, а также полностью использовать команды LPRINT, LIST "P:", PRINT #..."P:".

Пересылаются 7-битовые данные, поэтому можно применять непосредственно ASCII-коды принтера с десятичными значениями от 0 до 127. Остальные коды могут быть доступны после пересылки управляющих кодов. Например, для принтера Star NX-10 это будет команда LPRINT CHR\$(27);">", которая установит восьмой бит входных данных принтера на 1; возврат к кодам 0-127 произойдет после команды LPRINT CHR\$(27);"=".

Все надписи, сделанные в режиме "Inverse video", распечатываются обычным образом. Это происходит потому, что коды "inverse" используют восьмой бит ATASCII-кода, который маскируется управляющей программой. Благодаря такой перекодировке данных в листингах нет "нечитабельных" символов (ATASCII-коды и коды принтера с номерами 128-255 не соответствуют друг другу).

Несоответствие кода CR - возврат каретки - устраняется в управляющей программе (после кода CR всегда посылается LF).

Ниже приводится программа на Вейсике.

```

31999 REM ** HANDLER P: (PIA) **
32000 SU=0:RESTORE 32030
32010 READ X,Y:FOR I=X TO Y:READ Z:POKE I,Z:SU=SU+Z:NEXT I
32020 READ Z:IF Z<=1 OR SU<24279 THEN ? "Ошибка! Проверь
даные":END

```




```
32040 DRUX=USR(1536):? "OK":END
32050 DATA 1536,1781
32060 DATA 104,169,12,162,6,141,27,3,142,28,3,96,26,6,94,6,93,6,105,6
32070 DATA 69,6,93,6,76,239,6,173,2,211,41,251,141,2,211,169,255,141,0,211
32080 DATA 173,2,211,9,4,141,2,211,169,128,141,0,211,169,1,141,225,3,169,0
32090 DATA 141,222,2,141,29,208,32,70,6,96,169,83,141,2,3,141,10,3,172,225
32100 DATA 3,192,1,208,6,169,30,141,20,3,96,169,128,168,96,32,75,253,169,0
32110 DATA 141,225,3,76,229,6,201,27,208,13,9,128,141,224,3,160,0,140,226,3
32120 DATA 76,171,6,9,128,141,0,211,141,224,3,188,65,3,132,33,160,0,140,226
32130 DATA 3,201,155,208,26,169,13,9,128,141,224,3,141,0,211,32,171,6,169,10
32140 DATA 9,128,141,0,211,141,224,3,141,226,3,173,16,208,201,1,240,249,120,173
32150 DATA 224,3,41,127,141,0,211,173,224,3,141,0,211,88,173,226,3,201,10,208
32160 DATA 32,165,64,141,0,3,165,33,141,1,3,174,2,3,224,83,208,2,169,64
32170 DATA 141,3,3,173,20,3,141,6,3,169,155,162,0,160,1,140,3,3,96,169
32180 DATA 30,141,20,3,96,0,-1
```

После ввода и запуска программы с помощью команды RUN управляющий модуль принтера (handler) располагается на шестой странице памяти. После использования RESET необходимо выполнить команду X=USR(1536), чтобы система опять "увидела" управляющий модуль. Теперь остается только в соответствии со схемой подсоединить разъемы и принтер.

Предлагаем вашему вниманию еще одну программу, позволяющую генерировать так называемый BOOT-загрузчик, который запишет модуль управляющей программы на шестую страницу памяти, а затем приведет систему в состояние готовности к загрузке следующей программы типа BOOT (ею может быть, например, текстовый процессор SPEED SCRIPT). Программа написана на языке Turbo Basic XL версия 1.5:

```
31998 REM TURBO BASIC XL Генератор версии BOOT программы
      управления принтера Centronics
32000 SU=0:RESTORE 32050
32010 READ X,Y:FOR I=X TO Y:READ Z:POKE I,Z:SU=SU+Z:NEXT I
32020 READ READ Z:IF Z=0 OR SU=27423 THEN ? "Ошибка! Проверь
      данные":END
32030 OPEN #3,8,128,"C":BPUT #3,19000,279:CLOSE #3
32040 ? "Готово!"
32050 DATA 19000,19279
32060 DATA 0,3,228,5,234,5,169,60,141,2,211,169,0,133,8,169,1,133,9,141
32070 DATA 233,3,141,234,3,76,245,6,104,169,12,162,6,141,27,3,142,28,3,96
32080 DATA 26,6,94,6,93,6,105,6,69,6,93,6,76,228,6,173,2,211,41,251
32090 DATA 141,2,211,169,255,141,0,211,173,2,211,9,4,141,2,211,169,128,141,0
32100 DATA 211,169,1,141,225,3,169,0,141,222,2,141,29,208,32,70,6,96,169,83
32110 DATA 141,2,3,141,10,3,172,225,3,192,1,208,6,169,30,141,20,3,96,169
32120 DATA 128,168,96,32,75,255,169,0,141,225,3,76,229,6,201,27,208,13,9,128
32130 DATA 141,224,3,160,0,140,226,3,76,171,6,9,128,141,0,211,141,224,3,188
32140 DATA 65,3,132,33,160,0,140,226,3,201,155,208,26,169,13,9,128,141,224,3
32150 DATA 141,0,211,32,171,6,169,10,9,128,141,0,211,141,224,3,141,226,3,173,16
32160 DATA 208,201,1,240,249,120,173,224,3,41,127,141,0,211,173,224,3,141,0,211
32170 DATA 88,173,226,3,201,10,208,32,165,64,141,0,3,165,33,141,1,3,174
32180 DATA 2,3,224,83,208,2,169,64,141,3,3,173,20,3,141,6,3,169,155,162,0
32190 DATA 160,1,140,3,3,96,169,30,141
32200 DATA 20,3,96,32,1,6,32,124,198,0,-1
```

После ввода программы с клавиатуры и запуска командой RUN вы услышите два коротких звуковых сигнала. Нажмите RETURN, чтобы записать на кассету handler-загрузчик.

Для того чтобы загрузить редактор SPEED SCRIPT вместе с модулем принтера, надо включить компьютер, нажав клавиши OPTION и START; загрузить handler и после звукового сигнала загрузить SPEED SCRIPT, нажав на клавишу RETURN.

Текстовый процессор будет "сотрудничать" с принтером через порт A PIA. Однако существует одно ограничение: во время работы с редактором нельзя использовать клавишу RESET, так как операционная система компьютера Atari перестанет "видеть" модуль управляющей программы последовательного разъема. Представленное здесь решение с успехом было проверено на одном из популярных принтеров фирмы STAR - NX-10.

Для любопытных читателей приводим сделанную на Ассемблере запись модуля управляющей программы, составленную в формате ASSEMBLER/EDITOR.

```
10 ;Handler CENTRONICS
20 ;с использованием PIA
30      *=$600
40 PORTA      = $D300
50 MEMO      = 992
60 MARK      = 993
65 BLOCK      = 994
66 TRIGO      = $D010
70 PSTER      = $D302
75 BEGIN      PLA
80      LDA #TABLE,$00FF      LOW
90      LDX #TABLE/256      HIGH
0100     STA $031B
0110     STX $031C
0115     RTS
0120 ;ВЕКТОР ТАБЛИЦЫ УСТАНОВЛЕН
0130 TABLE .BYTE OPEN-$00FF
0140 .BYTE OPEN-1/256
0150 .BYTE CLOSE-$00FF
0160 .BYTE CLOSE-1/256
0170 .BYTE POWR-$00FF
0180 .BYTE POWR-1/256
0190 .BYTE PUT-$00FF
0200 .BYTE PUT-1/256
0210 .BYTE STATUS-$00FF
0220 .BYTE STATUS-1/256
0230 .BYTE POWR-$00FF
0240 .BYTE POWR-1/256
0250 .BYTE $4C
0260 .BYTE PON-$00FF
0261 .BYTE PON/256
0270 OPEN   LDA PSTER
0280      AND #FB
0290      STA PSTER
0300      LDA #FF
0310      STA PORTA
0320      LDA PSTER
0330      ORA #4
0340      STA PSTER
0350      LDA #80
0360      STA PORTA      7 бит -> Вывод
0370      LDA #1
0380      STA MARK
0390      LDA #0
0400      STA $2DE
0410      STA $D01D
0420      JSR STATUS
0430      RTS
0440      LDA #53
0450      STA $302
0460      STA $30A
0470      LDY MARK
0480      CPY #1
0490      BNE L1
0500      LDA #1E
0510      STA $314
0520      RTS
0530      LDA #80
0540      TAY
0550      RTS
0560 POWR    JSR $FF4B
0570 CLOSE  LDA #0
0580      STA MARK
0590      JMP CL1
0600 PUT    CMP #1B
0610      BNE TU
0620      ORA #80
0630      STA MEMO
0640      LDY #0
0650      STY BLOCK
0660      JMP L0
0670      ORA #80
0680      STA PORTA
0690      STA MEMO
0700      LDY $0341,X
0710      STY $21
0720      LDY #0
0730      STY BLOCK
0740      CMP #9B
```


068F	D01A	0770	BNE L0	
0691	A90D	0780	LDA #13	CR
0693	0980	0782	ORA #80	
0695	8DE003	0790	STA MEMO	
0698	8D00D3	0800	STA PORTA	
069B	20AB06	0810	JSR L0	
069E	A90A	0820	LDA #10	LF
06A0	0980	0822	ORA #80	
06A2	8D00D3	0830	STA PORTA	
06A3	8DE003	0840	STA MEMO	
06A8	8DE203	0850	STA BLOCK	
06AB	AD10D0	0860 L0	LDA TRIGO	
06AE	C901	0870	CMP #1	
06B0	F0F9	0880	BEQ L0	
06B2	78	0890	SEI	
06B3	ADE003	0900	LDA MEMO	
06B6	297F	0910	AND #7F	STB=0
06B8	8D00D3	0920	STA PORTA	
06BB	ADE003	0930	LDA MEMO	
06BE	8D00D3	0940	STA PORTA	
06C1	58	0950	CLI	14micsec STB=1
06C2	ADE203	0980	LDA BLOCK	
06C3	C90A	0990	CMP #10	
06C7	D020	1000	BNE KON	
06C9	A540	1010	LDA \$40	
06CB	8D0003	1020	STA \$300	
06CE	A521	1030	LDA \$21	
06D0	8D0103	1040	STA \$301	
06D3	AE0203	1060	LDX \$302	
06D6	E053	1070	CPX #53	
06D8	D002	1080	BNE L4	
06DA	A940	1090	LDA #40	
06DC	8D0303	1100 L4	STA \$303	
06DF	AD1403	1110	LDA \$314	
06E2	8D0603	1120	STA \$306	
06E5	A99B	1130 CL1	LDA #9B	
06E7	A200	1140	LDX #0	
06E9	A001	1150 KON	LDY #1	
06EB	8C0303	1160	STY \$303	
06EE	60	1170	RTS	
06EF	A91E	1180 PON	LDA #1E	
06F1	8D1403	1190	STA \$314	
06F4	60	1200	RTS	
06F5	1210	.END		

РЕШЕНИЕ 2

Представленное выше решение имеет несомненный плюс: система работает. Конечно, у этого решения есть и некоторые недостатки, которые, однако, легко исправить. Так, эта программа не возвращается к стандартному режиму работы порта A (вход), что делает невозможным использование джойстика после завершения работы с принтером. Перестает действовать клавиша BREAK, что весьма неудобно, так как программы, управляющие обслуживанием принтера, обычно предусматривают для пользователя возможность прерывания работы. Второй недостаток вместе с "герметичным" циклом ожидания готовности к работе принтера приводит к "зависанию" системы в том случае, когда принтер не подключен.

Кроме того, программа составлена с большим разбросом. Например, строки 1010 ... 1090 никогда не будут выполнены, так как значение в ячейке BLOCK всегда отличается от 10. Нет никакой пользы и от установки многочисленных ячеек массива DCB (от \$300).

Также представляется излишним использование трех рабочих ячеек памяти - MEMO, MARK и BLOCK.

Безотносительные адреса в программе вместо удобных для чтения флажков позволяют судить о том, что некоторые ее фрагменты были переписаны из дисассемблера.

Переход к системной процедуре определения длины буфера под \$FF4B также не нужен.

Трудно угадать, для чего служит отдельная трактовка кода \$1B (ESC), проверяемого в начале процедуры PUT. Неправильная установка этого кода ухудшает возможность

использования команд принтера, которые в большинстве своем начинаются именно с кода ESC.

Поэтому я предлагаю другое программное решение - программу на Бейсике, загружающую и запускающую обслуживание принтера через порты джойстика.

```

10 REM ----- "Centronics" -----
20 FOR A=1536 TO 1656:READ X
30 POKE A,X:S=S+X:NEXT A
40 IF S=11515 THEN ? "Исправы":END
50 A=USR(1536)
100 DATA 104,169,12,141,27,3,169,6
110 DATA 141,28,3,96,45,6,38,6
120 DATA 44,6,80,6,53,6,44,6
130 DATA 96,169,56,141,2,211,140,0
140 DATA 211,169,60,141,2,211,96,160
150 DATA 0,32,25,6,200,96,160,235
160 DATA 32,25,6,140,0,211,160,96
170 DATA 140,28,2,164,17,240,12,172
180 DATA 28,2,240,10,172,16,208,208
190 DATA 242,200,96,160,128,96,160
200 DATA 138,96,201,153,208,9,169,13
210 DATA 32,94,6,48,209,169,10,32,54
220 DATA 6,48,202,9,128,133,66,141
230 DATA 0,211,41,127,141,0,211,9,128
240 DATA 141,0,211,136,132,66,200,96

```

Ниже представлен листинг Ассемблера "JWB" (автор Я. Вишневецкий). Те, кто пользуются другими ассемблерами, смогут легко заменить DTA A(x) на .WORD X, ORG на *, EQU на = или OPT ... на соответствующие опции ассемблирования.

0000	OPT %0001111	0001
0000		0002
0000	*****	0003
0000	*	0004
0000	* Handler принтера	0005
0000	*	0006
0000	* "CENTRONICS"	0007
0000	*	0008
0000	* с использованием PIA	0009
0000	*	0010
0000	*****	0011
0000		0012
0000		0013
009B	EOLN EQU \$9B	0014
0060	TIME EQU \$60	0015
0000		0016
0000		0017
0011	BRKV EQU \$11	0018
0042	CRITC EQU \$42	0019
021C	TIMER3 EQU \$21C	0020
031A	HATABS EQU \$31A	0021
D010	TRIGO EQU \$D010	0022
D300	PORTA EQU \$D300	0023
D302	PACTL EQU \$D302	0024
0000		0025
0000		0026
0000		0027
0000	ORG \$600	0028
0000		0029
0000		0030
0000		0031
0000	68 PLA	0032
0001	A90C LDA >TABL	0033
0003	8D1B03 STA HATABS+1	0034
0006	A906 LDA >TABL	0035
0008	8D1C03 STA HATABS+2	0036
000B	60 RTS	0037
000C		0038
000C		0039
000C	*-- handler table--	0040
000C		0041
000C	TABL EQU *	0042
000C	2D06 DTA A(OPEN-1)	0043



060E	2606	DTA A(CLOSE-1)	0044
0610	2C06	DTA A(RETU-1)	0045
0612	3006	DTA A(PUT-1)	0046
0614	3506	DTA A(STATUS-1)	0047
0616	2C06	DTA A(RETU-1)	0048
0618	60	RTS	0049
0619			0050
0619			0051
0619		*— fixport A ———	0052
0619			0053
0619	A938	SEPA LDA #38	0054
061B	8D02D3	STA PACTL	0055
061E	8C00D3	STV PORTA	0056
0621	A93C	LDA #3C	0057
0623	8D02D3	STA PACTL	0058
0626	60	RTS	0059
0627			0060
0627			0061
0627		*— close ———	0062
0627			0063
0627	A000	CLOSE LDY #0	0064
0629	201906	JSR SEPA	0065
062C	C8	INV	0066
062D	60	RETU RTS	0067
062E			0068
062E			0069
062E		*— open ———	0070
062E			0071
062E	A0FF	OPEN LDY #255 out	0072
0630	201906	JSR SEPA	0073
0633	8C00D3	STV PORTA	0074
0636		*	0075
0636			0076
0636			0077
0636		*— status ———	0078
0636			0079
0636	A060	STATUS LDY #TIME	0080
0638	8C1C02	STV TIMER3	0081
063B	A411	WAIT LDY BRKCV	0082
063D	F00C	BEQ BREAK	0083
063F	AC1C02	LDY TIMER3	0084
0642	F00A	BEQ TIMEOUT	0085
0644	AC10D0	LDY TRIGO	0086
0647	D0F2	BNE WAIT	0087
0649	C8	INV	0088
064A	60	RTS	0089
064B		*— BREAK key ———	0090
064B	A080	BREAK LDY #128	0091
064D	60	RTS	0092
064E		*— no response ———	0093
064E	A08A	TIMEOUT LDY #138	0094
0650	60	RTS	0095
0651			0096
0651			0097
0651		*— write byte ———	0098
0651			0099
0651	C99B	PUT CMP #EOLN	0100
0653	D009	BNE OUTB	0101
0655	A90D	LDA #13 CR	0102
0657	205E06	JSR OUTB	0103
065A	30D1	BMI RETU	0104
065C	A90A	LDA #10 LF	0105
065E	203606	OUTB JSR STATUS	0106
0661	30CA	BMI RETU	0107
0663	0980	ORA #100000000	0108
0665	8542	STA CRITIC	0109
0667	8D00D3	STA PORTA	0110
066A	297F	AND #01111111	0111
066C	8D00D3	STA PORTA	0112
066F	0980	ORA #100000000	0113
0671	8D00D3	STA PORTA	0114
0674	88	DEY	0115
0675	8442	STV CRITIC	0116
0677	CB	INV	0117
0678	60	RTS	0118
0679			0119
0679			0120

© Андрей Жуков

Русские буквы на Atari ST

Несмотря на очевидные достоинства домашних компьютеров семейства Atari ST (быстродействующий процессор Motorola 68000 с тактовой частотой 8 МГц, объем памяти от 512 Кбайт до 4 Мбайт, хранящаяся в ПЗУ многооконная графическая операционная система, мышь и т.п.), они до сих пор имеют довольно ограниченное распространение на советском компьютерном рынке из-за практически полного отсутствия программ, адаптированных к русскому языку. Русификация Atari ST осложняется также тем, что графические символы на экране дисплея при работе прикладных программ формируются обычно преобразованием стандартных резидентных шрифтов, записанных в ПЗУ, а в них русский шрифт не предусмотрен.

Стандартный знакогенератор Atari ST имеет три отдельные таблицы шрифтов, хранящиеся в ПЗУ по адресам:

\$FD3A50 - шрифт размером 6x6 пикселей;

\$FD412C - шрифт размером 8x8 пикселей;

\$FD5B88 - шрифт размером 8x16 пикселей.

Шрифт 6x6 используется стандартно для подписей к пиктограммам (пиктограмма - icon) - графический элемент, включающий в себя изображение, букву и надпись, например пиктограмма дисковода), шрифт 8x8 - для вывода текстов на цветной дисплей в малом (320x200 пикселей, 16 цветов) и среднем (640x200 пикселей, 4 цвета) разрешениях, а шрифт 8x16 применяется для вывода на черно-белый дисплей высокого (640x400 пикселей) разрешения (пиксел - точечный элемент экранного изображения). Хотя для конкретных приложений достаточно ограничиться русификацией только того шрифта, который применяется в используемой вами прикладной программе, полностью возможности вывода графического текста с символами кириллицы на экран можно реализовать лишь при изменении всех трех таблиц.

Каждая таблица шрифта состоит из заголовка, содержащего вспомогательную информацию, таблицы отступов символов, таблицы горизонтальных отступов и собственно таблицы символов. Заголовок таблицы шрифта, который автоматически загружается в ОЗУ при инициализации системы, имеет структуру, показанную в таблице:

Адрес	Содержимое ячеек
0 - 1	Номер шрифта (1 - системный шрифт)
2 - 3	Размеры шрифта в пикселах
4 - 35	Название шрифта
36 - 37	Минимальный ASCII-код в таблице шрифта
38 - 39	Максимальный ASCII-код в таблице шрифта
40 - 41	Расстояние от Base line до Top line
42 - 43	Расстояние от Base line до Ascent line
44 - 45	Расстояние от Base line до Half line
46 - 47	Расстояние от Base line до Descent line
48 - 49	Расстояние от Base line до Bottom line



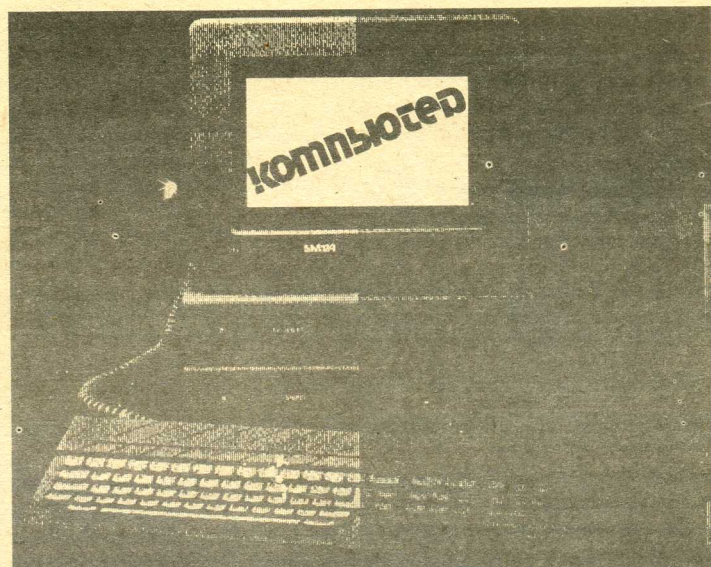
- 50 - 51 Ширина самого широкого символа
 52 - 53 Ширина самого широкого прямоугольника для символа
 54 - 55 Левый отступ при наклонном написании (курсив)
 56 - 57 Правый отступ при наклонном написании (курсив)
 58 - 59 Количество пикселей, на которое увеличивается ширина символа при жирном написании
 60 - 61 Высота линии подчеркивания в пикселах
 62 - 63 Маска для вывода на экран светлых символов (обычно 0x5555)
 64 - 65 Маска для вывода на экран курсива (обычно 0x5555)
 66 - 67 Флаги: бит 0 - если установлен, то системный шрифт;
 бит 1 - если установлен, то используется таблица горизонтальных отклонений;
 бит 2 - если установлен, то таблица символов задана в формате MS68000 (старший/младший байты), в противном случае - в формате Intel (младший/старший байты);
 бит 3 - если не установлен, то шрифт пропорциональный
 68 - 71 Указатель на начало таблицы горизонтальных отступов
 72 - 75 Указатель на начало таблицы отступов символов
 76 - 79 Указатель на начало таблицы символов
 80 - 81 Количество байтов в строке таблицы символов
 82 - 83 Количество строк в таблице символов
 84 - 87 Указатель на начало таблицы заголовка следующего шрифта

Таблица отступов символов содержит ASCII-коды символов по отношению к минимальному значению ASCII-кода в данном шрифте, поэтому в первой ячейке таблицы всегда содержится значение 0. Таблица горизонтальных отступов нужна для непропорционального шрифта; в ней для каждого символа задано положительное или отрицательное число, показывающее, на сколько пикселей нужно сдвинуть данный символ при выводе на экран.

Таблица символов построена следующим образом. Каждый символ кодируется построчно двоичным кодом (рис. 3), затем соответствующие строки всех символов в порядке их ASCII-кодов складываются в единую строку. После этого полученная строка разбивается побайтно, и каждый байт записывается в виде шестнадцатеричного числа. Для шрифтов 8x8 и 8x16 строка каждого символа занимает точно один байт, в то время как для шрифта 6x6 в трех байтах оказываются упакованы строки четырех символов. Отметим, что принятая для экранных шрифтов построчная кодировка (первая строка первого символа + первая строка второго символа + ... + вторая строка первого символа + ...) отличается от посимвольной кодировки (первая строка первого символа + вторая строка первого символа + ... + первая строка второго символа + ...), принятой в стандартных редакторах экранных и принтерных шрифтов для Atari ST (например, включенных в состав пакетов DEGAS Elite, Megafont ST, Stad и подобных). Именно поэтому шрифты, полученные при использовании таких стандартных редакторов, требуют дополнительной перекодировки. Программа для подобной перекодировки входит в состав пакета DEGAS Elite.

Создание таблицы символов, включающих необходимые символы кириллицы, - еще полдела. Для использования русских символов в прикладных программах нужны специальные драйверы экрана и клавиатуры, переключающие знакогенератор на таблицу загруженного шрифта. Для этой цели проще всего воспользоваться одной из имеющихся в СССР программ, например VIPFONT.PRГ или LVAFONT.ACC.

Программа VIPFONT.PRГ наиболее проста, но и наиболее ограничена по своим возможностям. Она перегружа-



ет только шрифт 8x8, включая таблицу заголовка шрифта, таблицу отступов символов, таблицу горизонтальных отступов и таблицу символов. Соответствующие данные для загрузки содержатся в файле VIPFONT (без расширения); организация файла соответствует описанной выше. К сожалению, программа не содержит драйвера клавиатуры, поэтому ввод символов кириллицы оказывается не очень удобным. Отметим, что в каталоге AUTO программа не работает; для автоматической загрузки шрифта при включении компьютера вам нужно воспользоваться специальными загрузчиками (например, GFA-Starter).

Гораздо удобнее программа LVAFONT.ACC. Во-первых, она позволяет перегружать сразу все три шрифта (6x6, 8x8 и 8x16), во-вторых, содержит подключаемый драйвер клавиатуры (LVARUSS.KBD) и, в-третьих, автоматически загружается при включении компьютера или перезагрузке операционной системы. Данные для каждого из загружаемых шрифтов (только таблица символов) содержатся в отдельных файлах (LVA0606.FNT, LVA0808.FNT и LVA0816.FNT). Переключение с английского шрифта на русский и обратно производится одновременным нажатием левой клавиши Shift и клавиши Alternate.

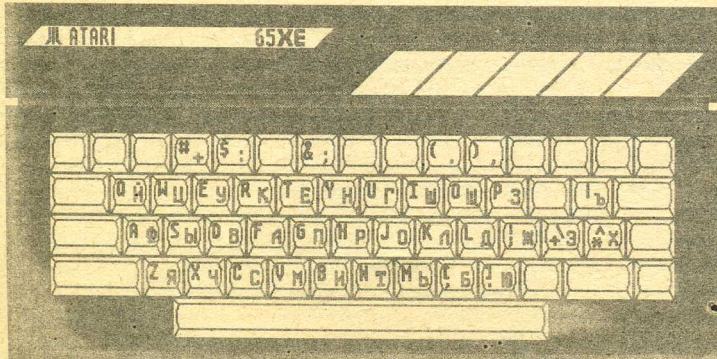
В заключение отметим, что расположение символов русского алфавита в таблицах шрифтов для Atari ST, имеющих в СССР, отличается от принятых для IBM-совместимых компьютеров. Это связано с тем, что наибольшее распространение компьютеры Atari ST получили в странах Западной Европы: алфавиты немецкого, французского, итальянского, испанского и других языков отличаются от английского наличием специальных дополнительных символов, занимающих в стандартной таблице Atari ST ASCII-коды со 128 по 167. Чтобы не нарушать вывод на экран системных сообщений (написанных на языке одной из европейских стран, где куплен компьютер), целесообразно присваивать символам кириллицы коды со 192 по 255 в алфавитном порядке (сначала прописные, затем строчные). Выделенные коды позволяют разместить все буквы русского алфавита (для строчной буквы "ё", если без нее не обойтись, может использоваться стандартный символ таблицы Atari ST с ASCII-кодом 137). Если символ с кодом 255 применять по какой-либо причине нежелательно (будьте внимательны при использовании кода 255: при непосредственной передаче информации на печать под этим кодом может быть расположена служебная команда принтера), строчной букве "я" можно присвоить один из кодов в интервале 168 - 191 (например, 145). Вместе с тем для выполнения операций сортировки русскоязычных текстов удобнее последовательное упорядочение букв с ASCII-кодами 195-255. Такое расположение букв русского алфавита, в частности, используется в русскоязычных орфографических словарях.

Первый превосходный

Действительно превосходный и действительно первый полноценный текстовый редактор для "маленького" Atari — The First XLEnt Word Processor. Его написал в 1986 г. Дэвид Кэстелл (David Castell) для фирмы XLEnt Software. На счету у этой фирмы и ранее было несколько отличных разработок для Atari XL/XE, прежде всего пакет, состоящий из 4 программ: Typesetter, Page Designer, Rubber Stamp и Megafont II+.

Чем замечателен XLEnt Word Processor с точки зрения русскоязычного пользователя? Прежде всего он позволяет без труда загрузить русский экранный шрифт. Как это сделать? Именно этим вопросом мы для начала и займемся.

Первым делом скопируйте содержимое дискеты с редактором The First XLEnt Word Processor на чистую дискету (этого правила следует придерживаться всегда: никаких переделок на программной дискете, только на копии). Теперь необходим соответствующий инструмент, т.е. программа, с помощью которой можно отредактировать экранный шрифт. Выберите ту, которая вам больше нравится. Я обычно пользуюсь программой Create-A-Font, написанной в 1986 г. Винсом Эрсегом (Vince Erceg). Советую сразу же "рисовать" символы русского алфавита там, где они действительно должны быть: "й" вместо "q", "ц" вместо "w", "у" вместо "e" и т.д. Одним словом, в как можно большем соответствии со стандартной русской клавиатурой ИЦУКЕН. Правда, есть одно небольшое затруднение — в русском алфавите букв больше, чем в латинском, поэтому от каких-то символов придется отказаться. Однако все необходимые буквы и знаки препинания на клавиатуре Atari XL/XE помещаются. Взгляните на рисунок. С левой стороны каждой клавиши находятся символы системного экранного шрифта, с правой — те, которыми я их заменил с помощью редактора шрифтов. На некоторых клавишах слева обозначено по два символа. В этом случае верхний из них заменен прописной, а нижний — строчной буквой. Исключение составляет "ъ" как единственная строчная буква, заменяющая верхний знак. На моей клавиатуре прописной твердый знак отсутствует по двум причинам: во-первых, он применяется чрезвычайно редко, во-вторых, я не хотел переносить символ "=". Однако при желании это так же можно сделать.



Когда экранный шрифт уже готов, надо записать его на копию дискеты с редактором. Файл со шрифтом можно записать либо с любым именем, либо с именем FONT.SYS. Разница в том, что если файл будет назван, например, RUSSKIJ.FNT, то после загрузки редактора его следует вызывать специальной командой, а если файлу присвоено имя FONT.SYS, то он загрузится автоматически. Команда загрузки шрифтового файла выглядит так: нажать одновременно клавиши [SHIFT], [CONTROL] и [S], в системную строку ввести имя файла, содержащего нужный шрифт (в нашем случае RUSSKIJ.FNT), и указать номер дисковода (вводя соответствующую цифру или подтверждая клавишей [RETURN] принимаемый по умолчанию номер 1).

У редактора The First XLEnt Word Processor есть еще одно интересное свойство — вместо графических символов (клавиша [CONTROL] и одна из 29 клавиш) можно ввести собственные символы (SPECIAL SYMBOLS). Эти символы вводятся также с помощью редактора экранного шрифта. Они должны быть записаны в том же шрифтовом файле. Работая с редактором, в любой момент можно проверить их расположение, нажимая клавишу [Help], а затем клавишу [4] (HELP SCREEN #4).

Заменяя системный экранный шрифт русским, вы столкнетесь с двумя неудобствами. The First XLEnt Word Processor выгодно отличается от других редакторов для Atari XL/XE тем, что в любой момент можно воспользоваться подсказкой-справочником (клавиши [Help] и [1] — [4], а также клавиша [Select]). Но после замены шрифта вся информация в справочнике превращается в абракадабру. Это первое неудобство. Однако из этого положения есть выход. На дискете может быть записано несколько шрифтовых файлов. Если вы работаете с русским шрифтом, а нужна подсказка, то можно известной командой ([SHIFT], [CONTROL], [S] и имя файла) загрузить латинский шрифт, прочитать необходимую информацию, снова загрузить русский шрифт и продолжить работу.

Теперь о втором неудобстве. В просмотрном режиме (имитация распечатки на экране, при которой вы можете увидеть действительный вид страницы) русских букв не будет, так как для этой операции требуется не загружаемый, а системный экранный шрифт.

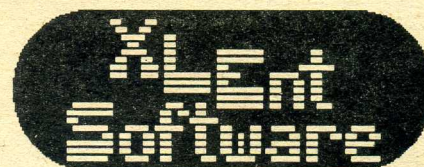
Теперь перейдем к не менее важному вопросу: как все это распечатать на бумаге? Прежде всего для этого необходим принтер, предусматривающий возможность программной загрузки шрифтов пользователя. Для полного успеха требуются еще две вещи — интерфейс Atari-Centronics и загрузочные файлы, содержащие нужные шрифты. Что касается интерфейса, то загляните на страницу 43, а загружаемыми шрифтами мы займемся здесь.

Среди форматирующих команд редактора The First XLEnt Word Processor есть команда sb:. Воспользовавшись ею, можно передать на принтер любой ASCII-код, который будет воспринят им как управляющая команда. Если конструкция принтера предусматривает возможность загрузки шрифта пользователя, то достаточно написать некоторую последовательность команд sb: с ASCII-кодами, соответствующими командам загрузки, и переслать ее на принтер так, как обычный текст, предназначенный для распечатки. Для разных принтеров требуются различные команды загрузки. В качестве примера рассмотрим метод написания файла, загружающего русский шрифт в режиме draft (черновая печать), для Epson-совместимого принтера NL-10 фирмы STAR MICRONICS.

```
sb27:sb120:sb0
sb27:sb58:sb0:sb0:sb0
sb27:sb38:sb0:sb65:sb139:sb120:sb132:sb0:sb132:sb122:
sb132:sb0:sb132:sb120:sb0:sb0
sb27:sb38:sb0:sb66:sb66:sb139:sb254:sb0:sb4:sb8:sb16:sb32:
sb64:sb0:sb254:sb0:sb0
sb27:sb38:sb0:sb68:sb68:sb139:sb130:sb124:sb130:sb16:sb130:sb16:
sb130:sb16:sb108:sb0:sb0
sb27:sb38:sb0:sb69:sb69:sb139:sb130:sb64:sb34:sb20:sb8:sb16:
sb32:sb64:sb128:sb0:sb0
sb27:sb38:sb0:sb70:sb70:sb139:sb30:sb32:sb72:sb128:sb8:sb128:
```


sb72:sb32:sb30:sb0:sb0
 sb27:sb38:sb0:sb71:sb139:sb254:sb0:sb128:sb0:sb128:sb0:
 sb128:sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb72:sb72:sb139:sb254:sb0:sb144:sb0:sb144:sb0:
 sb144:sb0:sb96:sb0:sb0
 sb27:sb38:sb0:sb73:sb73:sb139:sb254:sb0:sb2:sb0:sb254:sb0:sb2:
 sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb74:sb74:sb139:sb124:sb130:sb0:sb130:sb0:sb130:
 sb0:sb130:sb124:sb0:sb0
 sb27:sb38:sb0:sb75:sb75:sb139:sb2:sb4:sb136:sb112:sb128:sb0:
 sb128:sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb76:sb76:sb139:sb3:sb0:sb134:sb120:sb130:sb0:
 sb130:sb0:sb254:sb0:sb3
 sb27:sb38:sb0:sb77:sb77:sb139:sb0:sb254:sb0:sb18:sb0:sb18:sb0:
 sb18:sb0:sb12:sb0
 sb27:sb38:sb0:sb78:sb78:sb139:sb128:sb0:sb128:sb0:sb254:sb0:
 sb128:sb0:sb128:sb0:sb0
 sb27:sb38:sb0:sb79:sb79:sb139:sb254:sb0:sb2:sb0:sb254:sb0:sb2:
 sb0:sb254:sb0:sb3
 sb27:sb38:sb0:sb80:sb80:sb139:sb68:sb130:sb0:sb130:sb0:sb146:
 sb0:sb146:sb108:sb0:sb0
 sb27:sb38:sb0:sb81:sb81:sb139:sb254:sb0:sb4:sb72:sb144:sb32:
 sb64:sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb82:sb82:sb139:sb254:sb0:sb16:sb0:sb16:sb0:sb40:
 sb68:sb130:sb0:sb0
 sb27:sb38:sb0:sb83:sb83:sb139:sb254:sb0:sb18:sb0:sb18:sb0:sb18:
 sb12:sb0:sb254:sb0
 sb27:sb38:sb0:sb84:sb84:sb139:sb254:sb0:sb146:sb0:sb146:sb0:
 sb146:sb0:sb130:sb0:sb0
 sb27:sb38:sb0:sb85:sb85:sb139:sb254:sb0:sb128:sb0:sb128:sb0:
 sb128:sb0:sb192:sb0:sb0
 sb27:sb38:sb0:sb86:sb86:sb139:sb254:sb0:sb64:sb32:sb16:sb32:
 sb64:sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb87:sb87:sb139:sb254:sb0:sb2:sb0:sb2:sb0:sb2:
 sb0:sb254:sb0:sb3
 sb27:sb38:sb0:sb88:sb88:sb139:sb224:sb16:sb0:sb16:sb0:sb16:sb0:
 sb254:sb0:sb0:sb0
 sb27:sb38:sb0:sb89:sb89:sb139:sb254:sb0:sb16:sb0:sb16:sb0:sb16:
 sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb90:sb90:sb139:sb98:sb4:sb152:sb0:sb144:sb0:
 sb144:sb0:sb254:sb0:sb0
 sb27:sb38:sb0:sb97:sb97:sb11:sb56:sb0:sb68:sb0:sb254:sb0:sb68:
 sb0:sb56:sb0:sb0
 sb27:sb38:sb0:sb98:sb98:sb11:sb124:sb0:sb8:sb0:sb16:sb0:sb32:
 sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb100:sb100:sb11:sb124:sb0:sb84:sb0:sb84:sb0:
 sb84:sb40:sb0:sb0:sb0
 sb27:sb38:sb0:sb101:sb101:sb11:sb65:sb32:sb17:sb10:sb4:sb8:sb16:
 sb32:sb64:sb0:sb0
 sb27:sb38:sb0:sb102:sb102:sb11:sb8:sb20:sb64:sb20:sb64:sb20:
 sb64:sb56:sb4:sb0:sb0
 sb27:sb38:sb0:sb103:sb103:sb11:sb64:sb60:sb64:sb0:sb64:sb0:
 sb64:sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb104:sb104:sb11:sb127:sb0:sb68:sb0:sb68:sb0:sb68:
 sb56:sb0:sb0:sb0
 sb27:sb38:sb0:sb105:sb105:sb11:sb124:sb0:sb4:sb0:sb124:sb0:sb4:
 sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb106:sb106:sb11:sb56:sb68:sb0:sb68:sb0:sb68:sb0:
 sb68:sb56:sb0:sb0
 sb27:sb38:sb0:sb107:sb107:sb11:sb0:sb4:sb72:sb48:sb64:sb0:sb64:
 sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb108:sb108:sb11:sb6:sb72:sb52:sb64:sb4:sb64:sb4:
 sb120:sb6:sb0:sb0
 sb27:sb38:sb0:sb109:sb109:sb11:sb0:sb124:sb0:sb20:sb0:sb20:sb0:
 sb20:sb8:sb0:sb0
 sb27:sb38:sb0:sb110:sb110:sb11:sb64:sb0:sb64:sb0:sb124:sb0:sb64:
 sb0:sb64:sb0:sb0
 sb27:sb38:sb0:sb111:sb111:sb11:sb124:sb0:sb4:sb0:sb124:sb0:sb4:
 sb0:sb124:sb0:sb6
 sb27:sb38:sb0:sb112:sb112:sb11:sb40:sb68:sb0:sb84:sb0:sb84:sb0:
 sb84:sb40:sb0:sb0
 sb27:sb38:sb0:sb113:sb113:sb11:sb124:sb0:sb8:sb128:sb16:sb128:
 sb32:sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb114:sb114:sb11:sb0:sb124:sb0:sb16:sb0:sb16:sb0:
 sb40:sb68:sb0:sb0
 sb27:sb38:sb0:sb115:sb115:sb11:sb124:sb0:sb36:sb0:sb36:sb24:sb0:

sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb116:sb116:sb11:sb56:sb68:sb16:sb68:sb16:sb68:
 sb16:sb68:sb48:sb0:sb0
 sb27:sb38:sb0:sb117:sb117:sb11:sb0:sb124:sb0:sb64:sb0:sb64:sb0:
 sb96:sb0:sb0:sb0
 sb27:sb38:sb0:sb118:sb118:sb11:sb124:sb0:sb32:sb16:sb8:sb16:
 sb32:sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb119:sb119:sb11:sb120:sb4:sb0:sb4:sb0:sb4:sb0:
 sb124:sb0:sb6:sb0
 sb27:sb38:sb0:sb120:sb120:sb11:sb96:sb16:sb0:sb16:sb0:sb16:sb0:
 sb124:sb0:sb0:sb0
 sb27:sb38:sb0:sb121:sb121:sb11:sb124:sb0:sb16:sb0:sb16:sb0:sb16:
 sb0:sb124:sb0:sb0
 sb27:sb38:sb0:sb122:sb122:sb11:sb32:sb84:sb8:sb80:sb0:sb80:sb0:
 sb124:sb0:sb0:sb0
 sb27:sb38:sb0:sb58:sb58:sb139:sb130:sb108:sb16:sb0:sb254:sb0:
 sb16:sb108:sb130:sb0:sb0
 sb27:sb38:sb0:sb59:sb59:sb11:sb68:sb40:sb16:sb0:sb124:sb0:sb16:
 sb40:sb68:sb0:sb0
 sb27:sb38:sb0:sb92:sb92:sb139:sb68:sb0:sb130:sb0:sb146:sb0:
 sb146:sb68:sb56:sb0:sb0
 sb27:sb38:sb0:sb43:sb43:sb11:sb68:sb0:sb68:sb16:sb68:sb16:sb68:
 sb56:sb0:sb0:sb0
 sb27:sb38:sb0:sb94:sb94:sb139:sb130:sb68:sb40:sb16:sb0:sb16:
 sb40:sb68:sb130:sb0:sb0
 sb27:sb38:sb0:sb42:sb42:sb11:sb68:sb40:sb16:sb0:sb16:sb40:sb68:
 sb0:sb0:sb0:sb0
 sb27:sb38:sb0:sb91:sb91:sb139:sb254:sb0:sb146:sb0:sb146:sb0:
 sb146:sb0:sb12:sb0:sb0
 sb27:sb38:sb0:sb44:sb44:sb11:sb124:sb0:sb84:sb0:sb84:sb0:sb84:
 sb0:sb8:sb0:sb0
 sb27:sb38:sb0:sb93:sb93:sb139:sb254:sb0:sb16:sb0:sb124:sb0:
 sb130:sb0:sb130:sb0:sb124
 sb27:sb38:sb0:sb46:sb46:sb11:sb124:sb0:sb16:sb0:sb56:sb0:sb68:
 sb0:sb68:sb56:sb0
 sb27:sb38:sb0:sb124:sb124:sb11:sb64:sb0:sb124:sb0:sb20:sb0:sb20:
 sb0:sb8:sb0:sb0
 sb27:sb38:sb0:sb41:sb41:sb11:sb0:sb0:sb13:sb0:sb14:sb0:sb0:sb0:
 sb0:sb0:sb0
 sb27:sb38:sb0:sb40:sb40:sb11:sb0:sb0:sb12:sb0:sb12:sb0:sb0:sb0:
 sb0:sb0:sb0
 sb27:sb38:sb0:sb38:sb38:sb11:sb0:sb0:sb109:sb0:sb110:sb0:sb0:
 sb0:sb0:sb0:sb0
 sb27:sb38:sb0:sb36:sb36:sb139:sb0:sb0:sb54:sb0:sb54:sb0:sb0:sb
 0: sb0:sb0:sb0
 sb27:sb38:sb0:sb35:sb35:sb139:sb16:sb0:sb16:sb0:sb124:sb0:sb16:
 sb0:sb16:sb0:sb0
 sb27:sb37:sb49:sb0



Все форматирующие команды нашего редактора должны быть введены в отдельных строках-абзацах, начинающихся символом  (одновременное нажатие клавиш [Option] и [F]) и кончающихся символом "конец строки" (EOL, ATASCII 155), т.е. нажатием [RETURN]. Приведенный здесь файл разбит на 69 абзацев. Первый абзац содержит последовательность ASCII-кодов, которая выключает режим NLQ (высококачественной печати), перевода принтера в режим draft. Второй абзац - последовательность, являющаяся командой копирования стандартного шрифта принтера из его ПЗУ в ОЗУ. Абзацы с 3-го по 68-й содержат команды замены стандартных символов новыми. Таких абзацев 66, так как в них описываются все буквы русского алфавита (за исключением "Ъ", о чем было сказано выше, а также "С" и "с", которые имеют тот же вид, что латинские "с" и "С"), а также точка, запятая, двоеточие, точка с запятой и плюс, которые надо было перенести на новое место.





Каждый из этих абзацев построен в соответствии с требованиями принтера: первые три числа являются фактической командой загрузки, четвертая и пятая указывают номер символа, который будет заменен новым, шестая информирует принтер о том, которая (верхняя или нижняя) из 9 игловок головки не будет применяться, а остальные 11 описывают вводимый символ. Если вас интересует, какой символ вводится в данном абзаце, взгляните на четвертую и пятую величины, потом в таблицу символов принтера, а потом на рисунок клавиатуры. Например, в пятом абзаце на 4-м и 5-м местах вы найдете величину 68, которая соответствует прописной букве "D". На клавиатуре видно, что вместо "D" введена русская буква "В".

Последний абзац файла - это команда переключения принтера на загруженный шрифт. Из этого следует, что можно загрузить свой шрифт, потом переслать на принтер команду возврата к стандартному шрифту, а потом снова включить загруженный шрифт. Если вы собираетесь работать с принтером NL-10 или LC-10/LC-15, то загрузите ваш (и мой) любимый The First XLEnt Word Processor, перепишите файл, который мы анализировали, и запишите его на дискете как обычный текстовый файл. Если же у вас другой принтер, то сначала надо ознакомиться с инструкцией, в которой описываются команды загрузки, а затем по тому же принципу написать новый файл.

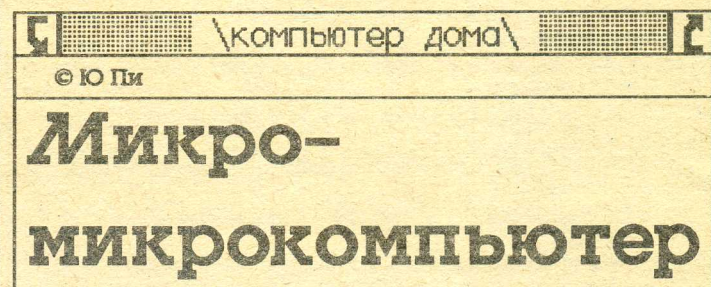
Точно так же можно написать файл для загрузки шрифта в режиме NLQ. Однако он будет гораздо длиннее, так как описание каждого символа при высококачественной печати состоит не из 11, а из 46 чисел. Кроме того, есть целый ряд принтеров, которые дают возможность загрузки пользовательского шрифта только в режиме draft (например, принтер KX-P1080i фирмы PANASONIC). Описанный метод загрузки шрифта весьма удобен, так как загрузку можно выполнять в любой момент на уровне редактора текста. При "внешнем" файле это приходится делать на уровне ОС или встроенного Бейсик-интерпретатора.

Для полноты картины вернемся еще к символам, которые можно ввести вместо графических символов (SPECIAL SYMBOLS). Как получить их на экране, мы уже обсудили. А как их распечатать? Сначала их надо спроектировать в соответствии с требованиями данной модели принтера. Затем символы следует ввести в существующий загрузочный файл, не забывая о необходимости продумать коды, под которые они будут загружаться, или же написать новый файл, на который можно будет переключиться в случае надобности (последний вариант не всегда удобен, поскольку XLEnt не дает возможности переключить шрифт только на 1 символ). До сих пор все было просто, но ведь почти все графические символы (кроме тех, которые имеют коды 96 и 123) находятся в начале таблицы ATASCII с кодами ниже 32. Одним словом, им надо приписать коды, которые приемлемы для принтера. Автор программы подумал и об этом. Если сразу после начала загрузки редактора с дискеты вы нажмете и придержите клавишу пробела, то на экране появится меню. Передвиньте подсветку на Printer Driver Construction и нажимайте [RETURN]. Таким образом вы загрузите модуль, в котором можно будет, во-первых, приписать каждому из 29 специальных экранных символов код любого символа, который принтер может отобразить на бумаге, во-вторых, приписать им так называемые маски, т.е. коды символов, которые будут появляться вместо них в просмотрном режиме (80 символов в строке). Результаты вашей работы с этим модулем запишите на дискету как файл с именем PRINTSET.SYS.

В завершение я хочу отметить, что загрузочный файл написан с учетом предложенной (см. рис.) клавиатуры. Если вы пожелаете изменить расположение символов на клавиатуре, то необходимо будет в загрузочный файл ввести соответствующие изменения. И еще одно замечание: способ написания загрузочного файла был обусловлен возможностями принтера NL-10, который позволяет загрузить пользовательский шрифт только в ту часть таблицы символов, которая находится между кодами 32 и 127.

□ □ □

The First XLEnt Word Processor - удобный и надежный инструмент для обработки текстов на Atari XL/XE. Для повышения эффективности работы с этой программой можно применять целый ряд приемов и "трюков". Если читатели выразят заинтересованность, то мы вернемся к этому редактору на страницах нашего журнала.



Московский завод "Ангстрем" любезно предоставил редакции "Компьютера" для описания первый советский карманный микрокомпьютер "Электроника МК 85".

Сначала о том, что видно на первый взгляд. Микрокомпьютер МК 85 настолько миниатюрен, что даже не верится, что это действительно ЭВМ. И все-таки это именно так. МК 85 является самым настоящим компьютером со встроенным Бейсиком и предназначен для составления программ, реализующих научные или инженерные расчеты. Однако добавим, что он может неплохо служить и для обучения основам программирования на Бейсике. Работа на МК 85 достаточно проста, а цена по нынешним временам весьма демократична (145 руб.), чтобы эта модель могла стать бестселлером для компьютерного обучения.

Машина (или, скорее, машинка) весит всего 150 г. Питание - 4 элемента СЦ 0,18 (для походных условий) или блок внешнего питания (при работе дома). В памяти МК 85 может одновременно находиться 10 программ на Бейсике, каждой из которых присваивается обязательное название P0 ... P9. Разумеется, программы сохраняются в памяти также после выключения питания. Кроме того, они не будут утрачены в течение 15 минут даже в случае извлечения элементов питания с целью замены на новые.

Основные режимы работы микрокомпьютера МК 85 - режим калькулятора и режим компьютера. Это довольно условное деление, так как в действительности существуют режимы записи, отладки, расширения функциональных возможностей, повышенного быстродействия и т.д. Однако для наших целей будем придерживаться упрощенной классификации.

Для успешной работы с МК 85 следует прежде всего ознакомиться с алфавитно-цифровой клавиатурой, которая на первый взгляд выглядит достаточно сложно. Такое впечатление вызывает большое количество условных обозначений вокруг почти каждой из 54 клавиш. Дело в том, что клавиши в зависимости от выбранного режима могут выполнять разные функции: вводить латинские или русские буквы, другие символы ASCII, названия функций или операторов (все очень просто для того, кто знаком со Спектрумом). Однако надо сказать, что клавиатура описана таким образом, что навыками работы на ней можно овладеть очень быстро. С помощью соответствующих клавиш можно также передвинуть курсор влево или вправо, очистить дисплей, устранить символ из строки или "раздвинуть" существующий текст для введения нового символа.

Запись программы в память или ее вызов и запуск производятся путем нажатия некоторой последовательности клавиш. Все операции на клавиатуре отображаются на жидкокристаллическом 12-разрядном дисплее. В строку можно ввести 63 символа. После 11-го символа начинается смещение строки влево. В верхней части дисплея выводится информация о режиме работы устройства и свободной памяти, измеряемой шагами программы.

Программы на встроенном в микрокомпьютер МК 85 Бейсике записываются с номерами строк. В программе можно использовать 1221 шаг, однако надо помнить, что некоторые элементы программы, например, номера строк, занимают по 2 шага (все остальные - команды, символы и функции - по 1). Нажатие клавиши "EXE", обязательное в конце каждой строки программы, - это 1 шаг.

Чтобы записать программу в память, надо перейти в режим записи и ввести текст программы с клавиатуры. После этого можно в любой момент вызвать ее, просмотреть листинг, ввести необходимые изменения.

Набор операторов и функций встроенного Бейсика достаточно широк для составления вполне серьезных программ. В "Руководстве по эксплуатации" (60 стр.) при описании операторов приводится ряд примеров программ с обширными комментариями, что помогает быстро разобраться в них даже начинающему пользователю. Способы использования возможностей клавиатуры МК 85 также описаны достаточно полно. Правда, можно выразить ряд замечаний в отношении полиграфического уровня "Руководства", но это как раз та деталь, которую фирма-изготовитель может при желании изменить в любой момент.

Очень хорошее впечатление у меня остается от работы с МК 85 в режиме калькулятора. Во-первых, четко соблюдается приоритет операций: сначала вычисление функций (тригонометрических, извлечения корня, округления и т.п.), затем возведение в степень, умножение, деление и в конце сложение и вычитание. Во-вторых, компьютер "понимает" запись со скобками, что очень удобно, так как соответствует навыкам пользователей. В-третьих, можно использовать ячейки памяти. На практике это выглядит так, что сначала определенному буквенному символу присваи-

вается числовое значение, после чего все операции в режиме калькулятора выполняются с использованием этого символа. Кстати, если вы выключите компьютер, а на следующий день захотите продолжить расчеты, то можете вновь воспользоваться тем же символом с прежним значением: машина сохранит введенную вами информацию.

У микрокомпьютера "Электроника МК 85" имеется один существенный недостаток. Кроме блока внешнего питания к нему ничего нельзя подключить, а хотелось бы иметь возможность, по крайней мере, записи программ и данных во внешней памяти и распечатки результатов своей работы на принтере, не говоря уже о подключении "малыша" к большой машине (такие возможности есть у зарубежных аналогов, например у карманных ЭВМ фирмы CASIO или SHARP). Однако и здесь есть надежда на изменение к лучшему: представитель фирмы "Ангстрем" сказал мне, что единственной причиной отсутствия возможности подключения к МК 85 внешних устройств является то, что ни одна советская фирма пока соответствующих устройств не выпускает. Остается надеяться, что в будущем "Ангстрем" наладит сотрудничество с производителями микропериферии и в корпусе МК 85 появятся дополнительные разъемы.

В заключение еще два факта, относящихся к работе с этим микрокомпьютером. Моей 8-летней дочери достаточно было часа, чтобы запомнить, как вводятся различные символы, научиться работать в режиме калькулятора, вызывать и запускать программу. Для меня же приятным сюрпризом была состоящая из трех строк программа, помещенная в "Руководстве" при описании операторов DRAW и DRAWC и выводившая на дисплей график функции $y = \sin x$. Если попробуете - оцените сами.



Что выбрать?

Когда в доме появляется компьютер, первым языком программирования становится обычно Бейсик, "встроенный" в ПЗУ машины. Несмотря на все свои недостатки, он имеет важное достоинство - он прост в изучении. По мере роста ваших компьютерных знаний вы можете начать программировать на Паскале или на не менее популярном Си. Но возможно, что вы полюбите Бейсик и останетесь верны ему, стараясь лишь найти отвечающий вашим потребностям диалект. Владельцы "маленьких" Atari имеют довольно широкие возможности выбора. В данной статье попытаемся сравнить 6 наиболее популярных диалектов Бейсика.

Основные свойства

Совместимость. Иначе говоря, это - внутреннее сходство между разными диалектами. "Совместимость с Atari-Бейсиком" означает, что программа, написанная на стандартном диалекте Atari, будет работать под управлением интерпретаторов других диалектов, независимо от введенных в них модификаций и изменений.

Быстродействие. Этот термин не требует особых объяснений. Каждому знакомы действующие на нервы моменты ожидания, когда компьютер выполняет введенную команду или монотонные вычисления. Однажды я видел, как трое суток продолжались сложные геодезические вычисления, а владельца компьютера мечтал только об одном - чтобы не отключили электроэнергию. Не отключили и "старушка" Atari 800 XL не подвела.

Компиляция. Прежде чем компьютер выполнит программу, она должна быть доведена до приемлемого для машины вида, т.е. "переведена" на машинный код. Большинство диалектов Бейсика начинает "перевод" после команды RUN. Программа "переводится" построчно. Так действует интерпретатор. Компилятор же переводит сразу всю программу и записывает откомпилированную версию. Она выполняется в 10-20 раз быстрее.

"Runtime package". Программный модуль, позволяющий написанную вами программу использовать тому, у кого нет данной версии Бейсика. Модуль присоединяется к тексту программы.

Atari Basic - один из лучших диалектов Бейсика. Существуют 3 его варианта. Вариант "C" находится в ПЗУ компьютеров серии XE, а для более ранних моделей он доступен в виде расширяющих модулей (cartridge). Иногда встречается вариант "A", который в свое время был разработан для Atari 400 и 800. Эти машины не имели интерпретатора в ПЗУ, в связи с чем вариант "A" чаще всего встречается в виде cartridge. Вариант "B" существует как "встроенный" в Atari 800 XL. Atari Basic является интерпретатором, но для него разработан ряд компиляторов. Можно также воспользоваться компилятором из пакета Turbo Basic XL. Модуль "runtime package" не нужен, так как в настоящее время в ПЗУ каждого 8-разрядного Atari "зашит" Atari Basic.

Microsoft Basic II - это диалект стандартного Бейсика, доступного почти для всех ПК. Он хорош в том случае, если вы вынуждены написанные на Бейсике программы перевести из ПЭВМ типа Apple или IBM PC на Atari. Этот диалект чрезвычайно эффективен, так как использует аппаратные возможности Atari. К сожалению, для него не был разработан компилятор. Его достоинства немного уменьшает тот факт, что он не совместим с диалектом Atari Basic, занимает слишком много памяти и не очень удобен в работе. Модуль "runtime package" отсутствует.

Появление диалекта **Basic XL** резко изменило положение тех, кто программировал на этом языке. Это был инструмент для "серьезных" программистов. Полная совместимость с Atari Basic, приличное быстродействие (в 4-5 раз быстрее, чем Atari Basic), удобство пользования, хорошая диагностика ошибок, операторы, аналогичные командам дисковой операционной системы - все это вызвало чрезвычайно теплый прием со стороны "атаристов". И хотя он требует 16 Кбайт памяти, но благодаря переключению блоков памяти и тому, что он заменяет стандартно "встроенный" Atari Basic, доступная для программиста память практически не уменьшается. К сожалению, с этим связано также то, что он доступен только в виде внешнего модуля (cartridge). Его достоинства: возможность работы со строковыми выражениями и возможность программирования сложной графики, обслуживание ввода/вывода, удобные операторы управления программой, высокая скорость вычислений. Серьезным недостатком является отсутствие компилятора. Возможно использование модуля "runtime package".

Basic XE имеет все свойства диалекта Basic XL, но дополнен рядом функций и чрезвычайно быстрыми математическими процедурами. У этого интерпретатора быстродействие в 6 раз выше, чем у диалекта Atari Basic, у него улучшенная графика, имеется возможность структурного программирования с использованием локальных переменных, формальных и актуальных параметров. В компьютерах Atari 130 XE использует всю доступную память, что позволяет писать и выполнять "длинные" программы. Компилятор отсутствует. Совместим с Atari Basic. Не имеет собственного "runtime package", но можно применять модуль из пакета Basic XL, при условии, что используются только функции этого диалекта.

Advan Basic чрезвычайно удобен для составления любых программ. Без каких-либо сложностей позволяет работать с графикой типа Player/Missile, писать музыкальное сопровождение программы или другие звуковые эффекты. Лучше других диалектов использует возможности 8-разрядных Atari. На этом диалекте можно писать игровые программы (хотя создавался он не для этого). Advan Basic - эффективный вариант Бейсика с расширенными операторами управления программой, отличной системой обслуживания ввода/вывода и операциями на строковых переменных. Компилятор позволяет ускорить выполнение программы в 10-15 раз по сравнению с Atari Basic. Не совместим с этим диалектом. Есть "runtime package".

Turbo Basic XL занимает место между стандартным Бейсиком Atari и диалектом Basic XL. Он общедоступен (public domain), а его автором является программист из ФРГ Франк Островски (Frank Ostrowski), известный всем пользователям Atari ST как создатель диалекта GFA Basic. Turbo Basic XL совместим с Atari Basic, но в 3-4 раза быстрее его. Располагает развитыми операторами управления программой и обслуживания ввода/вывода. Использование компилятора ускоряет выполнение программы в 16-20 раз. Предназначен только для компьютеров серии XL/XE с ОЗУ 64 Кбайта. В связи с общедоступностью "runtime package" не нужен.

Добрые советы

Если вы только начинаете программировать, выберите Atari Basic в версии "C" или Turbo Basic XL - в зависимости от того, что хотите делать. Если вы часто модифицируете свои программы, то лучшим выходом будет Basic XL или XE, а если хотите использовать мультипликацию, писать музыкальный фон или создавать сложные рисунки, выберите Advan Basic. Для длинных программ с крупными массивами данных и для компьютера Atari 130 XE возьмите Basic XE. Однако каждый из них имеет как достоинства, так и недостатки. Искусство выбора состоит в том, чтобы достоинств было как можно больше, а недостатков - как можно меньше. Может быть, вам поможет таблица.

Перевод Анджея Поплавского

Сравнение некоторых характеристик диалектов Бейсика

Диалекты	Atari Basic	Microsoft Basic II	Basic XL	Basic XE	Advan Basic	Turbo Basic XL
Тип компьютера и память	любой	любой 48 KB	любой	XL/XE 64 KB	любой 48 KB	XL/XE 64 KB
Совместимость с Atari Basic	x	нет	да	да	нет	да
Runtime package	x	нет	да	да (только функции Basic XL)	да	x
Компилятор	да	нет	нет	нет	да	да
Редакция и отладка программы						
DELETE строки	нет	да	да	да	да	да
Автоматическая нумерация строк	нет	да	да	да	нет	нет
Ренумерация	нет	да	да	да	нет	да
Слежение	нет	да	да	да	нет	да
Операторы управления программой						
IF/THEN/ELSE	нет	да	да	да	да	да
WHILE/WEND	нет	нет	да	да	да	да
REPEAT/UNTIL	нет	нет	нет	нет	да	да
CASE	нет	нет	нет	нет	да	да
PAUSE/WAIT	нет	да	нет	нет	да	да
Имена подпрограмм, процедур	нет	да	да	да	да	да
Ввод/вывод						
Каталог	нет	нет	да	да	да	да
DELETE filename	нет	да	да	да	да	да
LOCK/UNLOCK	нет	да	да	да	да	да
Двоичные LOAD/SAVE	нет	нет	да	да	да	да
INPUT	нет	да	да	да	да	да
PRINT USING	нет	да	да	да	отлично	нет
Строковые выражения						
Максимальная	память	128 байт	память	память	256 байт	память
Автомат. определение размерности массивов	нет	да	да	да	да	нет
Массивы строковых	нет	да	да	да	да	нет
Матрицы строковых	нет	нет	нет	нет	да	нет
LEFT\$/MID\$/RIGHT\$	нет	да	да	да	да	нет
Функции памяти						
PEEK/POKE	нет	нет	да	да	да	да
Блок MOVE	нет	да	да	да	нет	да
Присвоение значения блоку	нет	нет	нет	нет	нет	да
130XE - вся память	нет	нет	нет	да	нет	нет
Графика/звук						
Расширенная графика	нет	нет	нет	нет	хорошо	отлично
Графика P/M	нет	нет	хорошо	хорошо	отлично	нет
Расширенный звук	нет	хорошо	нет	нет	отлично	хорошо
Числа						
Быстрые мат. операции	нет	нет	нет	да	нет	нет
Целые	нет	да	нет	нет	да	нет
Шестнадцатеричные	нет	да	да	да	да	да
Двоичные	нет	нет	нет	нет	да	нет
Булевы операции	нет	AND/OR/XOR/NOT	AND/OR/XOR	AND/OR	AND/OR/XOR	AND/OR/XOR

x - смотри текст

Рисуем вместе с компьютером

Создание картинок на экране дисплея зачастую требует от программиста тонкого вкуса, художественных способностей и разносторонних знаний. Последнее обусловлено многообразием материала, который может стать основой будущей картинки. Таким материалом могут быть закономерности, действующие в живой природе, математические зависимости, простейшие геометрические фигуры или сложные пространственные тела необычной формы и многое другое - полный перечень необозрим.

Не ставя перед собой целью охватить все возможные подходы, мы иллюстрируем только некоторые, наиболее часто используемые.

Простая геометрия

Один из самых незамысловатых способов построения картинок основан на использовании элементарных геометрических фигур - отрезков прямых, окружностей, квадратов, треугольников и т.п. Построенное изображение часто напоминает детские рисунки или картинки для игры в мозаику (рис. 1). Программировать такие картинки несложно, если, конечно, в языке есть средства для работы с графическими примитивами (так часто называют простейшие геометрические фигуры, о которых идет речь), однако требуется предварительная кропотливая работа по определению размеров фигур и их размещению на экране.

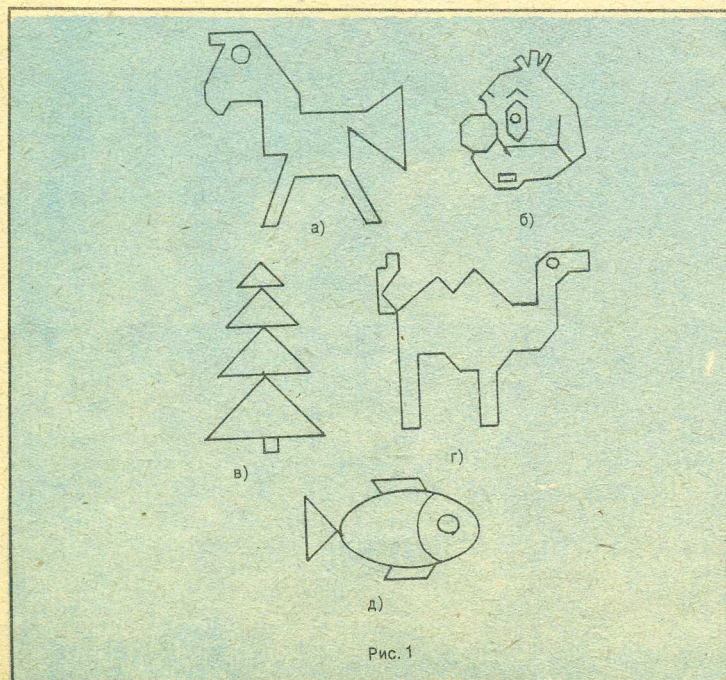


Рис. 1

Программа "Колодец" строит изображение из 30 вложенных квадратов. Длина стороны каждого следующего квадрата отличается от длины стороны каждого предыдущего на одну и ту же величину. Центр фигуры располагается в середине экрана, а сама она напоминает глубокий колодец, в который заглядываешь сверху.

```
10 REM Колодец
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
40 X=160: Y=100 'координаты центра фигуры
45 REM Строим тридцать вложенных квадратов
50 FOR I=1 TO 30
60 XL=X-I*3: YL=Y-I*3: XR=X+I*3: YR=Y+I*3
80 LINE (XL, YL)-(XR, YR), 1, B
90 NEXT
```

С помощью программы "Кораблик" на экране можно получить рис. 2.

```
10 REM Кораблик
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
40 COLOR 1,1
50 LINE (80,150)-(220,150),1: LINE -(200,175),1 'рисуем
60 LINE -(100,175),1: LINE -(80,150),1 'лодку
70 LINE (160,50)-(160,145),3 'рисуем
80 LINE -(90,145),3: LINE -(160,50),3 'парус
90 LINE (140,30)-(160,30),2: LINE -(160,45),2 'рисуем
100 LINE -(140,45),2: LINE -(150,37),2 'флаг
110 LINE -(140,30),2
120 PAINT (155,35),2,2 'закрашиваем флаг
130 PAINT (140,110),3,3 'закрашиваем парус
140 PAINT (180,160),1,1 'закрашиваем лодку
```

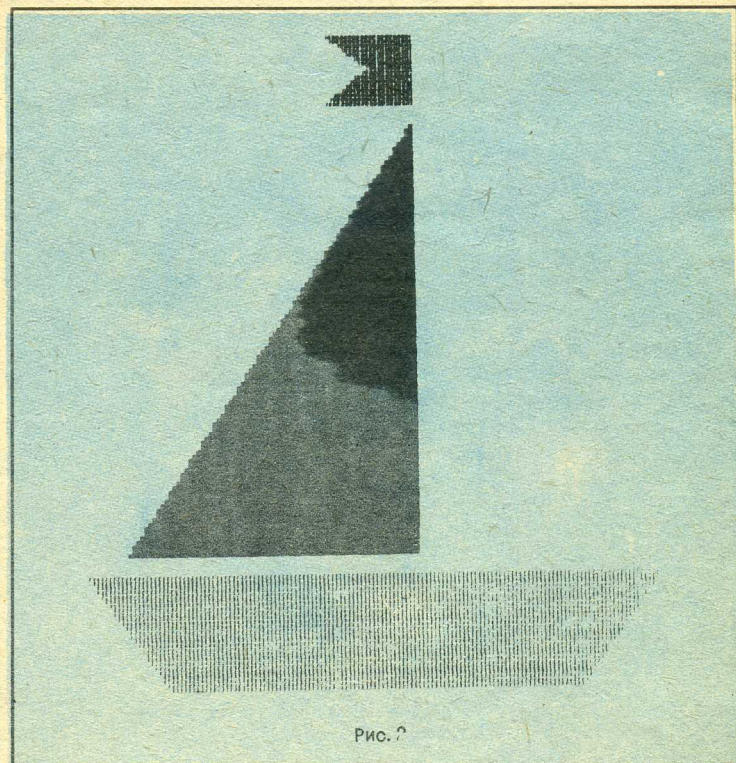


Рис. 2

Дело случая

Много интересных картинок может быть построено с помощью датчика случайных чисел. Изображение, созданное с использованием случайных величин (определяющих, в частности, элементы изображения, их цвета, размеры и расположение на экране), обладает эффектом непредсказуемости, неповторимости.

В программе "Смесь" изображение формируется случайным размещением на экране отрезков и окружностей произвольных размеров. Датчик случайных чисел применяется для выбора координат концов и цвета очередного отрезка (строка 50), а также для выбора радиуса, цвета и координат центра очередной окружности (строка 60).


```

10 REM Смесь
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
35 REM Строим двадцать отрезков и окружностей
40 FOR I=1 TO 20
45 X1=320*RND(1): Y1=200*RND(1): X2=320*RND(1): Y2=200*RND(1)
50 LINE (X1,Y1) - (X2,Y2), INT(3*RND(1))+1
60 CIRCLE (320*RND(1), 200*RND(1)), 100*RND(1), INT(3*RND(1))+1
70 NEXT

```

Программа "Конфетти" строит изображение из разнообразных символов, которые окрашиваются в разные цвета и произвольно разбрасываются по экрану.

```

10 REM Конфетти
20 REM IBM PC BASICA
30 WIDTH 80: SCREEN 0: KEY OFF: CLS
35 REM Выводим на экран 1000 символов
40 FOR I=1 TO 1000
50 C=INT(RND(1)*16): X=INT(RND(1)*79)+1
70 Y=INT(RND(1)*23)+1: A$=CHR$(INT(RND(1)*224)+32)
90 LOCATE Y,X: COLOR C,0: PRINT A$
120 NEXT

```

Датчик случайных чисел применяется в программе для выбора очередного символа, его цвета и положения на экране. Для тех, у кого вызвал недоумение способ выбора символа, заметим, что результат обращения $\text{INT}(224*\text{RND}(1))+32$ к функции RND (датчику случайных чисел) есть число из диапазона 32..255. Вы, вероятно, знаете, что символы, используемые в компьютере, кодируются числами от 0 до 255. Исключение из рассмотрения в программе символов с кодами от 0 до 31 вызвано тем, что эти символы являются управляющими - они выполняют некоторые специальные действия, связанные, например, с управлением дисплеем, звукогенератором или печатающим устройством. Вывод таких символов оператором PRINT может повлечь результат, неуместный в данной программе, например звуковой сигнал или очистку экрана.

"Калейдоскоп"

Симметрия, прочно вошедшая в нашу жизнь, проникла и в наши программы: большое число созданных к настоящему времени дисплейных картинок - это симметричные изображения. Фигуры, узоры, орнаменты и паркеты, формируемые на экране компьютера, часто имеют несколько осей симметрии.

Изображение, которое строится на экране программой "Калейдоскоп", напоминает узор в детской игрушке. Изображение имеет две оси симметрии - невидимые прямые AB и CD , делящие экран на четыре части (рис. 3,а). Для формирования изображения сначала выбирается точка в левом верхнем углу экрана (на рис. 3,а соответствующая область заштрихована). Выбор выполняется с помощью датчика случайных чисел. Полученная таким способом точка затем отражается относительно прямых AB и CD так, как показано на рис. 3,а.

```

10 REM Калейдоскоп
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
35 REM Строим 4*1500 точек
40 FOR I=1 TO 1500
50 X=INT(160*RND(1)): Y=INT(100*RND(1))
70 PSET(X,Y),2 'на экране появилась точка
75 REM Следующие 3 точки получены ее отражением
80 PSET(319-X,Y),2: PSET(319-X,199-Y),2
100 PSET(X,199-Y),2
110 NEXT

```

Изображение станет более красочным, если при его построении воспользоваться разными цветами. Номер цвета можно задать в цикле или выбрать с помощью датчика случайных чисел, например, так:

```

65 C=INT(3*RND(1))+1: PSET(X,Y),C: PSET(319-X,Y),C
90 PSET(319-X,199-Y),C: PSET(X,199-Y),C

```

Изображение можно немного усложнить, если взять не две, а, скажем, четыре оси симметрии. На рис. 3,б показаны предлагаемые оси симметрии, область, в которой с помощью датчика случайных чисел будет выбираться исход-

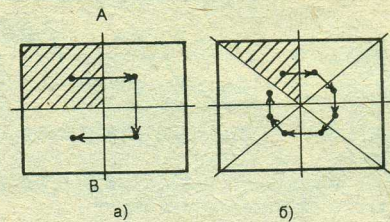


Рис. 3

ная точка (область заштрихована) и возможная последовательность построения симметричных точек (изображена стрелками).

Простота и красота

Большое число разнообразных картинок может быть построено на основе математических зависимостей. Созданным таким образом фигуры часто вызывают восхищение своей красотой и необычностью.

Программа "Павлин" фиксирует на экране положения отрезка, концы которого перемещаются следующим образом. Один из концов движется по горизонтальной прямой, проходящей через центр экрана; для вычисления координат другого конца используются тригонометрические функции $\sin$ и $\cos$. Результирующее изображение напоминает сказочную птицу (рис. 4).

```

10 REM Павлин
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
40 FOR X1=0 TO 319
50 X2=120+100*SIN(X1/30): Y2= 90+100*COS(X1/30)
70 LINE(X1,100)-(X2,Y2)
80 NEXT

```

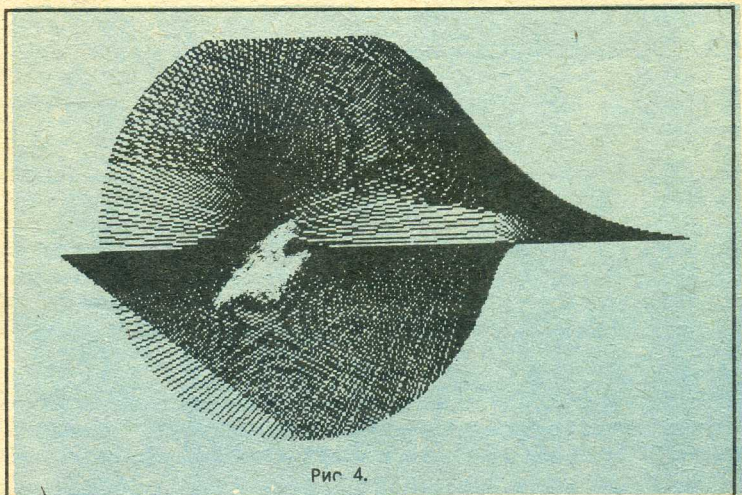


Рис. 4.

Попробуйте использовать для вычисления координат концов отрезка другие зависимости, и вы получите не менее интересные картинки.

Узор, который рисует программа "Кружева" (рис. 5), образован следующим образом. На экране строятся вершины правильного восемнадцатиугольника, центр которого совпадает с центром экрана. Каждая из восемнадцати вершин соединяется отрезками со всеми другими вершинами. Координаты вершин задаются формулами

$$X_i = X_c + R \cos(2\pi i/n);$$

$$Y_i = Y_c + R \sin(2\pi i/n), \quad i=1, \dots, 18,$$

где i - номер вершины, R - радиус окружности, описанной около многоугольника, X_c , Y_c - координаты его центра. Во избежание повторного вычерчивания отрезков между одними и теми же вершинами каждая из них соединяется только с вершинами, имеющими больший номер.



```

10 REM Кругева
20 REM IBM PC BASICA
30 SCREEN 1: KEY OFF: CLS
40 N=18 'число вершин правильного многоугольника
50 OPTION BASE 1
60 DIM X(N), Y(N) 'координаты вершин многоугольника
70 R=99 'радиус описанной окружности
80 DT=2*3.1416/N
90 T=0
95 REM Вычисление координат вершин многоугольника
100 FOR I=1 TO N
110 T=T+DT
120 X(I)=R*COS(T): Y(I)=R*SIN(T)
130 NEXT
135 REM Соединение вершин многоугольника
140 FOR I=1 TO N-1
150 FOR J=I+1 TO N
160 LINE (X(I),Y(I))-(X(J),Y(J))
170 NEXT
180 NEXT

```

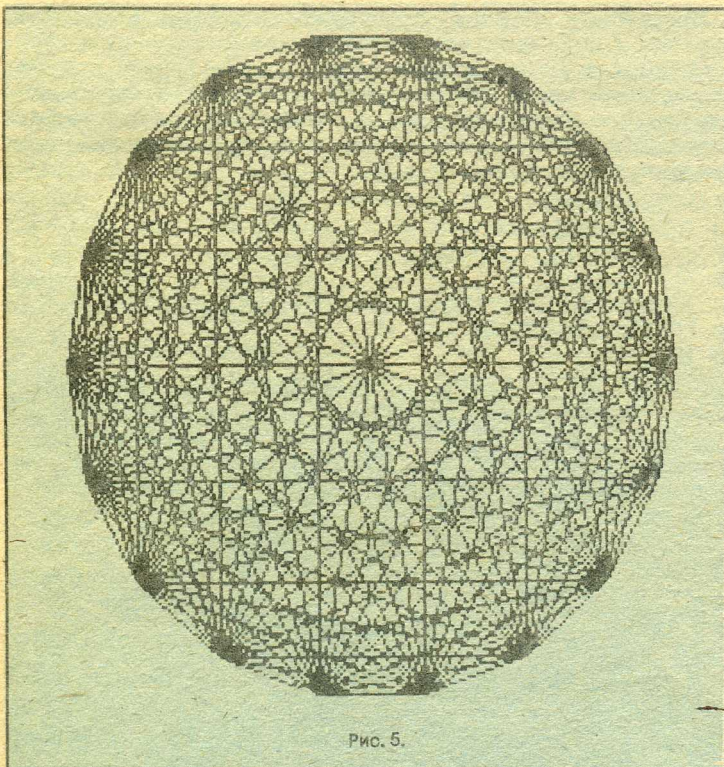


Рис. 5.

Измените количество вершин многоугольника (переменная N в строке 40), и вы увидите, как изменится узор.

Рисунок, вычерчиваемый программой "Убегающий квадрат" (рис. 6), на первый взгляд кажется достаточно сложным, но при более внимательном рассмотрении оказывается, что он образован вложенными квадратами. Вершины каждого следующего квадрата делят стороны предыдущего в заданном отношении μ . Таким образом, квадраты не только становятся все меньше и меньше, но и поворачиваются на некоторый угол.

```

10 REM Убегающий квадрат
20 REM IBM PC BASICA
30 DIM X(3), Y(3), XD(3), YD(3)
40 SCREEN 1: KEY OFF: CLS
50 R=100 'длина стороны внешнего квадрата
60 XL=100: YU=50 'координаты его левого верхнего угла
70 N=30 'число квадратов на рисунке
80 X(0)=XL: X(1)=XL+R: X(2)=XL+R: X(3)=XL
90 Y(0)=YU+R: Y(1)=YU+R: Y(2)=YU: Y(3)=YU
100 SMU=.08: RMU=1-SMU
105 REM Строим N квадратов
110 FOR I=1 TO N
115 REM Вычисляем координаты вершин очередного квадрата
120 FOR J=0 TO 3

```

```

130 XD(J)=RMU*X(J)+SMU*X((J+1) mod 4)
140 YD(J)=RMU*Y(J)+SMU*Y((J+1) mod 4)
150 NEXT
155 REM Строим очередной квадрат
160 FOR J=0 TO 3
170 LINE (X(J),Y(J))-(X((J+1) mod 4), Y((J+1) mod 4))
180 NEXT
190 FOR J=0 TO 3
200 X(J)=XD(J)
210 Y(J)=YD(J)
220 NEXT
230 NEXT

```

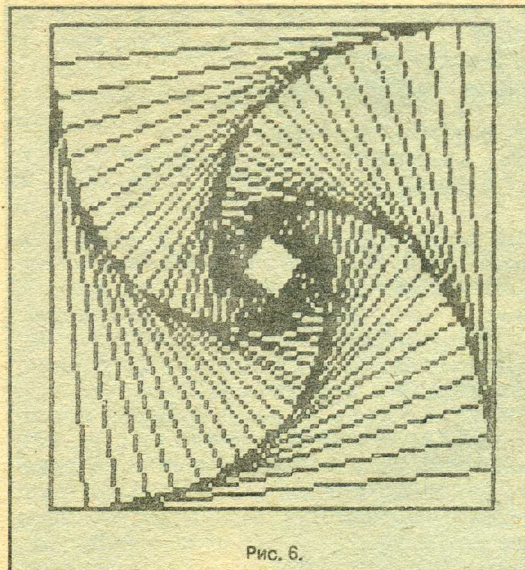


Рис. 6.

Исходными данными для программы являются координаты левого верхнего угла внешнего квадрата (100,50), длина его стороны (сто точек экрана), количество квадратов, которые должны быть построены (30) и значение $\mu = 0,8$. Для определения координат вершин очередного квадрата используются со-

отношения, позволяющие по известным координатам концов отрезка (X_1, Y_1) , (X_2, Y_2) и заданному отношению (μ), в котором некоторая точка делит этот отрезок, определить координаты (X, Y) этой точки:

$$X = X_1 + \mu (X_2 - X_1);$$

$$Y = Y_1 + \mu (Y_2 - Y_1).$$

Картинка, которая создается программой "Геометрический узор" (рис. 7), состоит из нескольких узоров, подобных построенному в предыдущем примере. Эффект достигается определенным чередованием узоров, в каждом из которых квадраты вращаются либо по, либо против часовой стрелки: узоры, стоящие в строке и столбце, номера которых являются числами одной четности, образованы вращением квадрата по часовой стрелке; узоры, стоящие в строке и столбце, номера которых являются числами разной четности, образованы вращением квадрата против часовой стрелки.

```

1 REM Геометрический узор
2 REM IBM PC BASICA
3 DIM X(3), Y(3), XD(3), YD(3)
4 SCREEN 1: KEY OFF: CLS
5 R=60: XL=10: YU=17
6 REM Рисуем две строки по три узора в каждой
7 FOR K=1 TO 2
8 FOR L=1 TO 3
9 X(0)=XL: X(1)=XL+R: X(2)=XL+R: X(3)=XL
10 REM Определяем направление вращения
11 IF K MOD 2 = 0 AND L MOD 2 = 0 THEN 28
12 IF K MOD 2 = 1 AND L MOD 2 = 1 THEN 28
13 Y(0)=YU+R: Y(1)=YU+R: Y(2)=YU: Y(3)=YU
14 GOTO 40
15 Y(0)=YU: Y(1)=YU: Y(2)=YU+R: Y(3)=YU+R
16 SMU=.08: RMU=1-SMU
17 REM Вращаем квадрат
18 FOR I=1 TO 21
19 FOR J=0 TO 3
20 XD(J)=RMU*X(J)+SMU*X((J+1) mod 4)
21 YD(J)=RMU*Y(J)+SMU*Y((J+1) mod 4)
22 NEXT

```



```

110 FOR J=0 TO 3
120 LINE (X(J),Y(J))-(X((J+1) mod 4), Y((J+1) mod 4))
130 NEXT
150 FOR J=0 TO 3
160 X(J)=XD(J)
170 Y(J)=YD(J)
180 NEXT
190 NEXT
200 XL=XL+R
210 NEXT
220 XL=10: YU=YU+R
230 NEXT

```

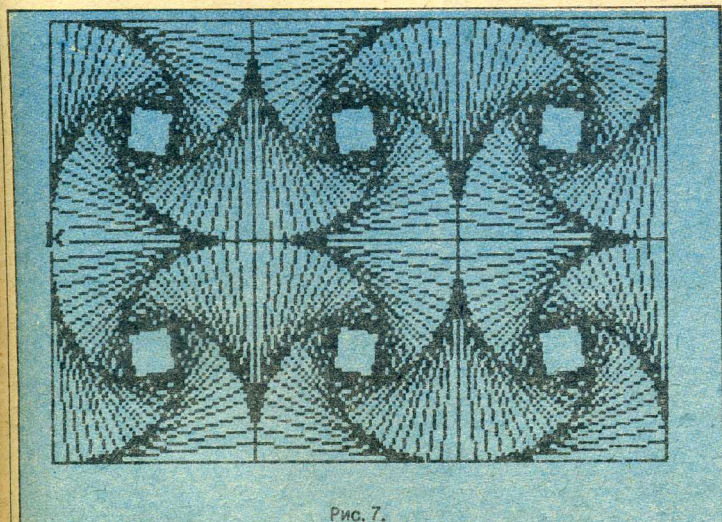


Рис. 7.

Узоры такого рода часто строятся авторами дисплейных картинок на основе не только квадратов, но и других многоугольников, например треугольников или пятиугольников. Дополнительной проблемой, которая возникает при таком построении, является расположение отдельных многоугольников на экране.

Немного физики и математики

Программа "Пять кривых" строит на экране кривые, хорошо известные любителям математики: спираль Архимеда, улитку Паскаля, кардиоиду, трилистник и четырехлистник. Уравнения кривых в полярных координатах имеют следующий вид (напомним, что полярные координаты ρ , ϕ точки M на плоскости - это расстояние $\rho=OM$ от фиксированной точки O (полюса) до точки M и угол $\phi=POM$ между OM и полярной осью (полупрямой) OP (рис. 8):

спираль Архимеда - $\rho=a\phi$, $a>0$;
улитка Паскаля - $\rho=a\cos\phi + 1$, $a>1, 1>0$;
кардиоиды - $\rho=a(1+\cos\phi)$, $a>0$;
трилистник - $\rho=a\cos 3\phi$, $a>0$;
четырёхлистник - $\rho=a\cos 2\phi$, $a>0$.

Угол ρ изменяется в программе от 0 до 2π , пересчет полярных координат в декартовы выполняется по формулам $x=\rho\cos\phi$, $y=\rho\sin\phi$.

```

1 REM Пять кривых
2 REM IBM PC BASICA
5 REM Выбор кривой
10 PRINT "Какую кривую вы хотите построить:"
20 PRINT " 1 - спираль Архимеда;"
30 PRINT " 2 - улитку Паскаля;"
40 PRINT " 3 - кардиоиду;"
50 PRINT " 4 - трилистник;"
60 PRINT " 5 - четырехлистник;"
65 PRINT " 6 - выход;"
70 INPUT K
75 IF K<1 OR K>6 THEN 10
77 IF K=6 THEN END
80 SCREEN 1: KEY OFF: CLS
85 REM Ввод параметра и построение кривой
90 INPUT "Задайте положительное число "; A
100 IF A<=0 THEN 90

```

```

110 DFI=1/A
120 ON K GOSUB 205,305,405,505,605
130 FOR FI=0 TO 6.28 STEP DFI
140 ON K GOSUB 250,350,450,550,650
150 X2=100+R*COS(FI): Y2=100-R*SIN(FI)
170 LINE (X1,Y1)-(X2,Y2): X1=X2: Y1=Y2
190 NEXT
193 FOR I=1 TO 500: NEXT
195 GOTO 10
200 REM Спираль Архимеда
205 X1=100: Y1=100: RETURN
250 R=A*FI: RETURN
300 REM Улитка Паскаля
305 L=A/2: X1=100+A+L: Y1=100: RETURN
350 R=L+A*COS(FI): RETURN
400 REM Кардиоиды
405 X1=100+A*2: Y1=100: RETURN
450 R=A*(1+COS(FI)): RETURN
500 REM Трилистник
505 X1=100+A: Y1=100: RETURN
550 R=A*COS(3*FI): RETURN
600 REM Четырёхлистник
605 X1=100+A: Y1=100: RETURN
650 R=A*COS(2*FI): RETURN

```

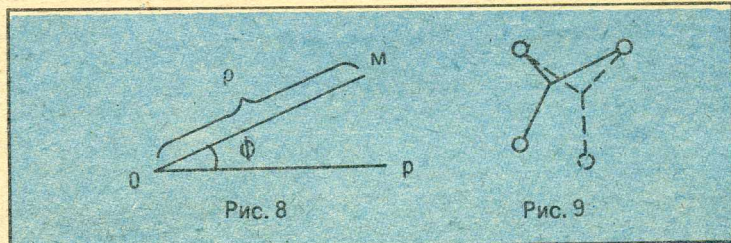


Рис. 8

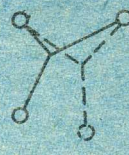


Рис. 9

Программа "Фигуры Лиссажу" строит на экране кривые, которые были названы именем французского физика, первым давшего их описание. Фигуры Лиссажу представляют собой результат сложения двух гармонических колебательных движений, совершаемых во взаимно перпендикулярных направлениях. Примером тела, участвующего в таких движениях, может служить маятник, изображенный на рис. 9.

(Напомним, что гармоническими называются такие колебания, при которых отклонение x тела от положения равновесия изменяется по закону $x=a\sin(\omega t+\phi)$, где a - амплитуда, ω - частота, ϕ - начальная фаза колебаний. Гармонические колебания совершают, например, математический маятник или напряжение в контуре электрической цепи.) Фигуры Лиссажу характеризуются уравнениями

$$x=a_1\sin(\omega_1 t+\phi_1);$$

$$y=a_2\sin(\omega_2 t+\phi_2)$$

и могут быть довольно сложными, особенно при близких частотах продольных и поперечных колебаний. На рис. 10 показана кривая, отвечающая уравнению:

$$x=\sin(2t);$$

$$y=\sin(7t+2)$$

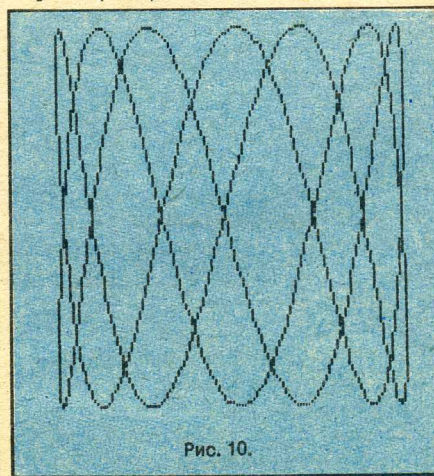


Рис. 10.

С помощью данной программы вы можете наблюдать влияние частот продольных и поперечных колебаний, а также начальной фазы продольных колебаний на форму фигур.



```

10 REM Фигуры Лиссажу
15 REM IBM PC BASICA
20 KEY OFF
50 WX=320: WY=200: XC=WX/2: YC=WY/2: C=1: MAX=WY/2-40
100 SCREEN 0: WIDTH 80
110 CLS: LOCATE 24,28: PRINT "F10 - в**од"
120 LOCATE 3,20
130 PRINT "Уравнения кривой Лиссажу имеют вид:"
140 PRINT TAB(26); "x=sin(M*t)"
150 PRINT TAB(26); "y=sin(N*(t+ALFA))"
160 PRINT: PRINT
170 INPUT "Частота колебаний по оси OX "; M
180 INPUT "Частота колебаний по оси OY "; N
210 INPUT "Начальная фаза колебаний по оси OY "; ALFA
220 LOCATE 18,20: PRINT "x=sin(",M;,"* t)"
230 LOCATE 19,20: PRINT "y=sin(",N;,"* t";
240 IF ALFA=0 THEN PRINT ")": GOTO 270
250 IF ALFA>0 THEN PRINT "+"; ELSE PRINT "-";
260 PRINT ABS(ALFA);")"
270 INPUT "Уравнения записаны верно? (Да/Нет)", Q$
280 IF LEFT$(Q$,1)="" AND LEFT$(Q$,1)="" THEN 110
1000 SCREEN 1: COLOR 0,1
1030 X1=XC: Y1=YC-SIN(N*ALFA)*MAX
1040 FOR T=0.01 TO 6.28318 STEP 0.01
1050 LOCATE 1,10: PRINT "t = "; PRINT USING "###";T;
1060 X2=XC+SIN(M*T)*MAX
1070 Y2=YC-SIN(N*(T+ALFA))*MAX
1080 LINE(X1,Y1)-(X2,Y2),C: X1=X2: Y1=Y2
1090 NEXT
1100 LOCATE 1,5: PRINT SPC(10);"КРИВАЯ ЛИССАЖУ"
1110 LOCATE 25,7: PRINT "Нажмите какую-нибудь клавишу";
1120 Q$=INKEY$: IF Q$="" THEN 1120
1130 GOTO 100

```

* Статья перепечатана из книги "Персональный компьютер в играх и задачах". - М.: Наука, 1988.



Музыкальная шкатулка *

Существует много музыкальных программ для маленького АТАРИ, в том числе довольно сложных. Есть даже специальные музыкальные редакторы, занимающие целую сторону дискеты. Однако, для того, чтобы заставить АТАРИ исполнить понравившееся Вам произведение, причем достаточно сложное, хватит и шести строк. Вот простая программа, которая позволяет это сделать:

```

10 TRAP 60: L=4
20 READ A,B,C,D
30 FOR I=1 TO L
40 SOUND 0,A,10,10: SOUND 1,B,10,6: SOUND
2,C,10,6: SOUND 3,D,10,6
50 NEXT I: GOTO 20
60 END

```

Сама музыка записывается в операторе DATA следующим образом.

Прежде всего выбирается наименьшая длительность нот, имеющихся в нотной записи понравившегося вам произведения, например, это может быть одна шестнадцатая (как у меня), или восьмушка, или четверть - важно только, чтобы это была самая короткая нота. Каждой такой длительности будет соответствовать одна строка с оператором DATA. В моей программе запись музыки начинается со строки 100:

```

100 DATA 53,144,182,217
102 DATA 0,0,0,0

```

В строке 100 записан аккорд из четырех нот длительностью в одну шестнадцатую (числа для нот вы можете взять из таблицы), далее в строке 102 следует короткая пауза (в одну шестнадцатую). А как же тогда заставить АТАРИ сыграть восьмушку или четверть? Очень просто для восьмушки придется повторить оператор DATA два раза, для четверти - четыре раза и т.д. Конечно, для записи музыки потребуется довольно много времени, но зато ошибки практически исключены - вы всегда легко найдете и исправите случайную ошибку: первая цифра - нота для первого голоса, вторая - для второго, третья - для третьего, а четвертая - для четвертого голоса (напомним, что у АТАРИ четыре звуковых канала). В том случае, если стоит ноль, значит в этом месте данный голос молчит. Вот и все! На с. 60 приводится начало листинга программы (вступление к танго). Попробуйте завершить программу самостоятельно. Таким образом, вы сможете стать настоящим компьютерным композитором.

Теперь о том, как работает программа. В строке 10 оператор TRAP сработает, когда иссякнут числа в операторе DATA; без него в этот момент программа бы остановилась, выдав сообщение об ошибке. Параметр L устанавливает

* Статья перепечатана из первого номера ежемесячного журнала "Club OK!" московского детского клуба "Компьютер". - М.: 1990. Ее автору - 15 лет.

темп исполнения произведения, если L уменьшить - темп музыки будет быстрее. Этот параметр используется в цикле (строки 30- 50). В Турбо Бейсике L возьмите равным 12. В строке 20 считываются четыре нотки из очередного оператора DATA, которые сейчас же исполняются в строке 40 операторами SOUND для четырех голосов. Обратите внимание, что громкость первого голоса (10) - мелодия - больше, чем громкость аккомпанирующих голосов (6). GOTO в

строке 50 заставляет все повторяться, пока не кончатся числа в операторе DATA. Если вы делаете музыкальную заставку к своей программе, то вместо оператора END в строке 60 оператором GOTO отправьте компьютер в начало вашей программы. Не забудьте, что последними в операторе DATA должны стоять четыре ноля (выключение звука).

Вперед, меломаны!



```

3 REM Musical program - TANGO -
4 REM Roma Makeev, Club "Computer", Moscow
5 TRAP 1300
10 TRAP 60:L=4
20 READ A,B,C,D
30 FOR I=1 TO L
40 SOUND 0,A,10,10:SOUND 1,B,10,6:
SOUND 2,C,10,6:SOUND 3,D,10,6
50 NEXT I:GOTO 20
60 END
100 DATA 53,144,182,217
102 DATA 0,0,0,0
104 DATA 53,144,182,217
106 DATA 0,0,0,0
108 DATA 0,0,0,0
110 DATA 0,0,0,0
112 DATA 91,0,0,0
114 DATA 72,0,0,0
116 DATA 47,0,0,0
118 DATA 47,0,0,0
120 DATA 53,0,0,0
122 DATA 53,0,0,0
124 DATA 72,0,0,0:126 DATA 72,0,0,0
128 DATA 91,0,0,0
130 DATA 91,0,0,0
132 DATA 53,136,162,193
134 DATA 0,0,0,0
136 DATA 53,136,162,193
138 DATA 0,0,0,0
    
```

```

140 DATA 0,0,0,0
142 DATA 0,0,0,0
144 DATA 0,0,0,0
146 DATA 68,0,0,0
148 DATA 47,0,0,0
150 DATA 47,0,0,0
152 DATA 53,0,0,0
154 DATA 53,0,0,0
156 DATA 68,0,0,0
158 DATA 68,0,0,0
160 DATA 81,0,0,0
162 DATA 81,0,0,0
164 DATA 60,136,153,182
166 DATA 0,0,0,0
168 DATA 60,136,153,182
170 DATA 60,136,153,182
172 DATA 60,136,153,182
174 DATA 0,0,0,0
176 DATA 68,136,153,182
178 DATA 0,0,0,0
180 DATA 60,136,153,182
182 DATA 60,136,153,182
184 DATA 60,136,153,182
186 DATA 0,0,0,0
188 DATA 68,136,153,182
190 DATA 68,136,153,182
192 DATA 68,136,153,182
194 DATA 68,136,153,182
196 DATA 72,144,162,230
    
```

ТАБЛИЦА ВЫСОТЫ МУЗЫКАЛЬНЫХ НОТ

Малая октава	до 251	до#/ 230	ре 217	ре#/ 204	ми 193	фа 182	фа#/ 173	соль 162	соль#/ 153	ля 144	ля#/ 136	си 128
Первая октава	до 126	до#/ 114	ре 108	ре#/ 102	ми 96	фа 91	фа#/ 85	соль 81	соль#/ 76	ля 72	ля#/ 68	си 64
Вторая октава	до 60	до#/ 57	ре 53	ре#/ 50	ми 47	фа 45	фа#/ 42	соль 40	соль#/ 38	ля 35	ля#/ 33	си 31
Третья октава	до 30	до#/ 28	ре 27	ре#/ 25	ми 23	фа 22	фа#/ 21	соль 20	соль#/ 18	ля 17	ля#/ 16	си 15
Четвертая октава	до 14	до#/ -	ре 13	ре#/ 12	ми 11	фа 10	фа#/ -	соль 9	соль#/ -	ля 8	ля#/ -	си 7
Пятая октава	до 6	до#/ -	ре 5	ре#/ -	ми 4	фа 3	фа#/ -	соль 2	соль#/ -	ля 1	ля#/ -	си -

Примечание: числа для 4 и 5 октав даны условно - звук немного фальшивый.

Г-н Юрий Парад, директор компании Oracle Corp. по Центральной Европе и СССР, сообщил редакции журнала "Компьютер", что Oracle Corp., известная как старейший разработчик технологии реляционных БД, контролирующей в настоящее время около половины мировых продаж в этом классе систем, выходит в ближайшее время на советский рынок.

Поздний старт фирмы в Союзе связан с тем, что предлагаемая теперь советским потребителям продукция до последнего времени подпадала под ограничения СОСОМ.

Компания намерена продавать в СССР сетевые реляционные СУБД Oracle для распределенной обработки данных в неоднородных системах (включающих компьютеры разных типов с различными ОС), со встроенным языком обработки запросов SQL, развитыми средствами для программистов и конечных пользователей. Система Oracle одинаково работает под множеством операционных систем (MVS, VMS, VM, UNIX, Vines, XENIX, MS-DOS, OS/2 и др.) на компьютерах всех классов — от mainframes с процессорами параллельной обработки до РСв. Распределенная архитектура системы обеспечивает коммуникационную прозрачность при работе объединенных в одной системе разнообразных по типу компьютеров. Мобильность Oracle позволяет пользователю разрабатывать СУБД на одном компьютере, например, персональном, а потом запускать ее в промышленную эксплуатацию на большой ЭВМ, не внося никаких изменений в исходные тексты программ.

Благодаря стандартизации SQL в ANSI и ISO Oracle совместима на уровне данных с DB2 (IBM) и dBaseIV (Ashton-Tate). Привлекательным (и модным) в глазах советских пользователей выглядит также реализация в Oracle CASE-технологии разработки прикладных систем.

В качестве дистрибьютера продукции фирмы в СССР выбрано малое предприятие R-Systems/Реляционные системы, возникшее на базе неформального клуба пользователей-энтузиастов ранних версий Oracle (несанкционированным образом привезенных в Союз) и Oracle-подобных систем типа KAPC, ПАЛЬМА, имеющих на ЭВМ советского производства (серий CM и EC).

Фирма намерена организовать в СССР ряд учебных центров, поддерживать консультативными услугами сеть пользователей Oracle, издавать книги и документацию на русском языке. Помимо этого всерьез рассматривается возможность продажи системы за рубли.

Заявки с неполным (неточным) адресом или адресом

"до востребования" не регистрируются.

Просим москвичей и ленинградцев оставлять предварительные заявки

непосредственно в

книжных магазинах,

распространяющих

научно-техническую

литературу: заказы на

высылку журнала

наложенным платежом из

Москвы и Ленинграда

не обслуживаются.

Отправку журнала

наложенным платежом

пока осуществляют два

магазина:

191 186 г. Ленинград,

Невский пр-т, 28.

Магазин N 1 "Дом книги";

101 000 г. Москва,

ул. Кирова, 6.

Магазин N 120

"Дом книги".

В остальных магазинах

можно оставлять лишь

предварительный

заказ для оповещения

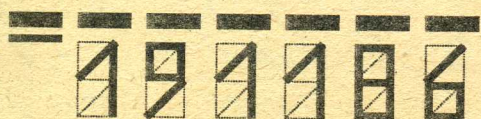
о возможности

непосредственной

покупки журнала.

ПОЧТОВАЯ КАРТОЧКА

Куда г. Ленинград
Невский пр., 28
Магазин N 1 "Дом книги"
 Кому Отдел "Книга - почтой"



Пишите индекс предприятия связи места назначения

Индекс предприятия связи и адрес
отправителя

(Полный адрес
читателя)

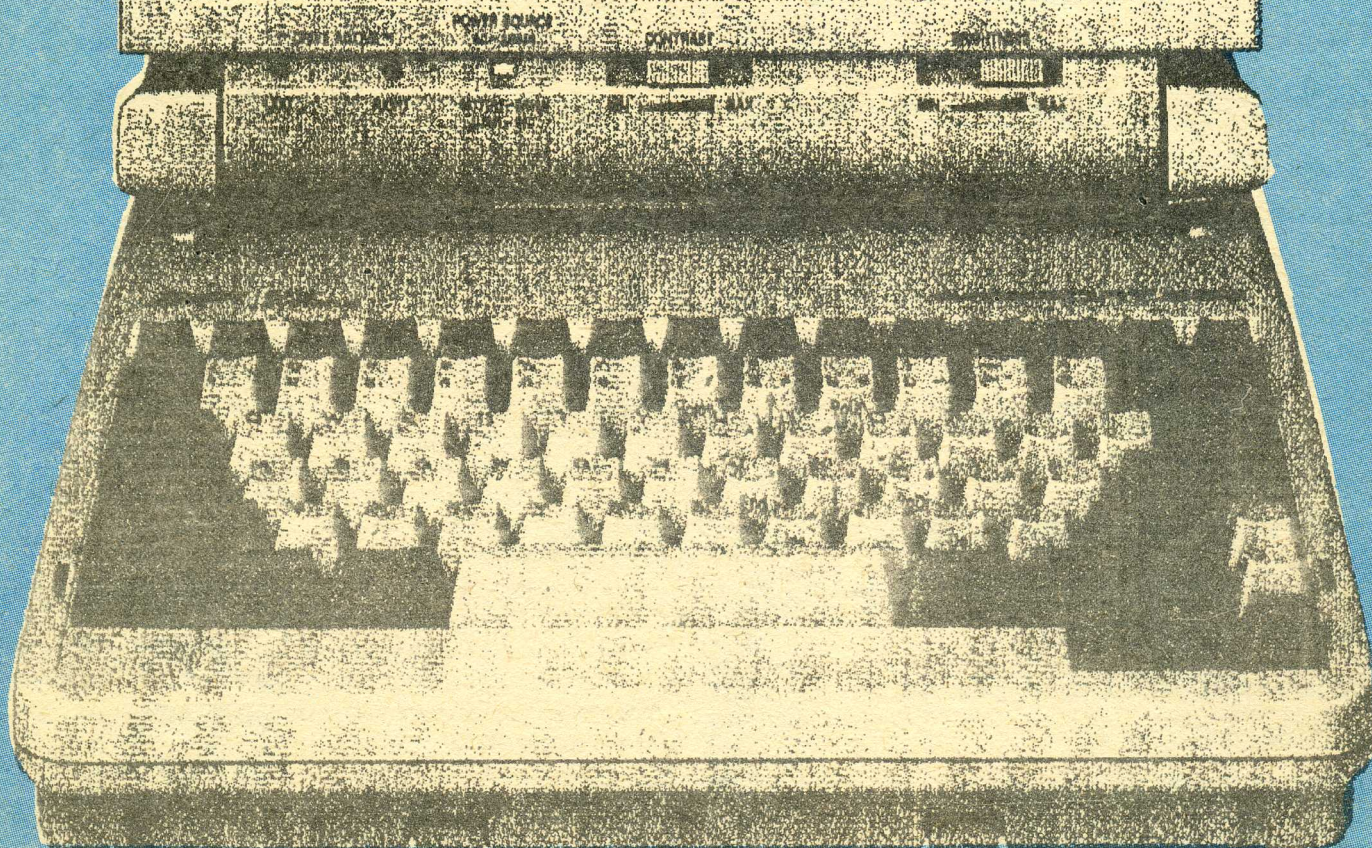
Просим выслать журнал
 "Компьютер" (издательство
 "Финансы и статистика")
 наложенным платежом в
 количестве экз., начиная с 1(4)
 номера за 1991 г. по адресу:

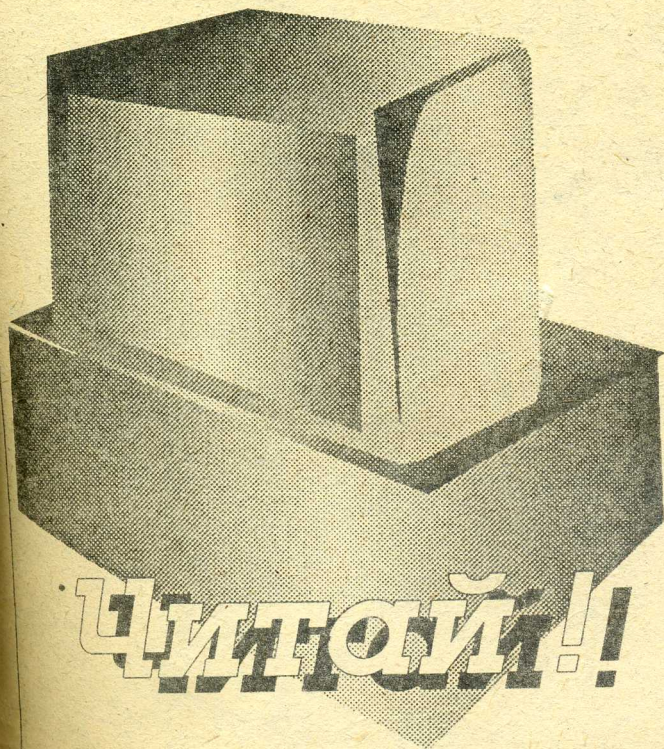
(полный адрес с индексом и
 указанием ф.и.о. получателя)

АНКЕТА

Участника конкурса

1. Возраст
2. Пол
3. Образование
4. Город село
5. Какой у Вас компьютер?
 - а. Атари
 - б. ZX Spectrum
 - в. IBM PC
 - г. Другой 8-разрядный (какой?)
- д. Нет





Следуя совету наших читателей, посетовавших на "не доставаемость" книг из ретроспективной библиографии, опубликованной в вып.1 "Компьютера", 1990 г., публикуем на сей раз список реально доступных изданий - только что изданных или издаваемых в I квартале 1991 г., - которые могут заинтересовать многих пользователей и программистов ПК.

Издательство "Мир" совместно со Всесоюзным объединением "Воким" и малым предприятием "Малип" продолжает издание серии микропамяток - компактных брошюр карманного формата. Заказывать их наложенным платежом рекомендуем в следующих магазинах:

121019 Москва, Проспект Калинина, 26, п/я 42. Книжный магазин N 200

191186 Ленинград, Невский проспект, 28, Дом книги

Вот какие издания из этой серии можно заказать:

1. Потоцкий В.К.

Работаем с системой Clipper

М.: Мир, 1990. - 95с. - 2 р. 20 к. 100000 экз.

«Рекомендуем заказывать в Ленинграде».

2. Морозова Н.В., Куликова Е.А.

Работаем со ScanGal

М.: Мир, 1990. - 94 с. 300000 экз.

ScanGal - пакет фирмы Хьюлетт Паккард для работы со сканером ScanJet и ScanJet Plus. Приводится его описание и некоторые принципы работы с подобными программами.

3. Белов В.А.

Работаем с dBASE III Plus.

М.: Мир, 1991. - 78 с. - 2 р. 30 к. 100000 экз.

Компактный справочник по командам, функциям и синтаксису популярной СУБД.

4. Головач В.И.

Работаем с SuperCalc 5.

М.: Мир, 1991. - 94 с. - 2 р. 100000 экз.

Справочник пользователя популярного табличного процессора для финансово-экономических расчетов и поддержания небольших табличных баз данных (в описываемой версии пакета имеется возможность создания трехмерных таблиц).

«Рекомендуем заказывать в Москве».

5. Морозова Н.В., Куликова Е.А.

Работаем с PaintBrush

М.: Мир, 1991. - 63 с. - 1 р. 50 к. 50000 экз.

Описан известный графический пакет (версия 2.0), весьма популярный у художников-оформителей и издателей.

«Рекомендуем заказывать в Москве».

6. Татарникова Л.В., Лазукова Т.Н., Оношко Д.Д.

Работаем с Word 5.0: в 2-х выпусках.

М.: Мир, 1991. - 187 с. - 4 р. (за комплект) 100000 экз.

Описаны приемы подготовки документов, оформление изданий, в том числе работа с иллюстрациями в популярнейшем текстовом процессоре.

Помимо этой серии отдельной книжкой выпущена также следующая работа:

Шумихин А., Шабанов А.

Подготовка текстов на ПЭВМ с помощью Word 5.0.

М.: Мир, 1990. - 4 р. 100000 экз.

Совместное предприятие "ICE" открыло для пользователей ПК библиотечку популярного журнала "Мир ПК". В ней изданы:

1. Безруков Н.Н.

Классификация компьютерных вирусов MS DOS и защита от них.

М.: СП "ICE", 1990. - 48 с. - 1 р. 50 к. 30000 экз.

2. Беляк А.И., Розенберг Д.Е.

Пользователю PC Tools - PC Shell.

М.: СП "ICE", 1990. - 64 с. - 2 р. 30000 экз.

3. Зильберман А.И.

Приобретая зарубежный персональный компьютер.

М.: СП "ICE", 1990. - 64 с. - 2 р. 20000 экз.

4. Масловский Е.К., Смирнов А.Ф., Кузьмин Ю.А.,

Теплицкий Л.А. Новые англо-русские термины по вычислительной технике.

М.: СП "ICE", 1990. - 48 с. - 1 р. 50 к. 30000 экз.

Заказы на приобретение этих книжек рекомендуем отправлять по адресу:

117922, Москва, Ленинский проспект, 15, ВО "Союзкнига".

Подборку подготовил Константин Коробов

Редакция **Компьютер**

и фирма KAREN
объявляют конкурс на разработку
совершенно новой, оригинальной
компьютерной игры.

Предложения участников конкурса
будут оцениваться в двух возрастных
группах

- до 12 лет и свыше 12 лет.

Авторы из младшей группы должны
представить следующие элементы игры:
описание "действительности",
в которой происходит действие;
цель игры и условия ее успешного завершения;
условия завершения каждого
этапа игры, если их несколько;
описание действий играющего;
список трудностей, которые надо преодолеть;
эскизы, иллюстрирующие игру.

Авторов из старшей группы мы
просим дополнительно представить:
систему счета очков;
описание последовательности ситуаций
игры или ее возможных разветвлений;
эскизы, иллюстрирующие графику игры;
описание действий компьютера, если
противником в игре является машина;
описание достоинств игры, то есть
элементов, которые могут привлечь играющих;
сравнение предлагаемого сценария с другими
играми и обоснование его превосходства.

Конкурсные работы просим направлять по адресу.

"KAREN"
01-573 Warszawa
ul. Podstaroscich 3
POLSKA

На конверте просим дописать "Конкурс".

Для авторов наиболее интересных
сценариев мы приготовили призы.

В младшей группе:
компьютер ZX Spectrum;
набор игр для ZX Spectrum;
бесплатная подписка на журнал "Компьютер".
В старшей группе:
компьютер Atari 130 XE;
дисквод LDW для Atari;
набор компьютерных игр.

Авторы, сценарии которых будут использованы при
разработке коммерческих вариантов игр, получат авторские
гонорары.

Всех участников просим вместе со сценарием игры прислать
заполненную анкету со страницы 62.
Конкурсные работы будут приниматься до 1 сентября 1991 г.
Результаты будут опубликованы в четвертом выпуске
"Компьютера" за 1991 г.

