«ИНФОРКОМ»

DRAFT VERSION 0.14

PDF version by NUK, km, Александр Кульнев
Если Вы хотите принять участие в обработке оставшихся статей, mailto:nuk@mail.ru.

ZX-РЕВЮ

1992

Выпуск 01-06

STALINGRAD	33[1]	Completed	by Александр Кульнев
FLIGHT SIMULATION	35	Completed	by Александр Кульнев
STAR RAIDERS 2	36	Completed	by Александр Кульнев
THE TRAIN	37	Completed	by Александр Кульнев
DEATH WISH 3	38	Completed	by Александр Кульнев
PROFESSIONAL TENNIS	77	Completed	by km
SNOOKER	77	Completed	by km
QUAZATRON	79	Completed	by km
CAPTAIN FIZZ	82	Completed	by km
FORUM	84	Completed	by km
ACADEMY	125	Completed	by km
SHERLOCK	129	Completed	by km
Наш конкурс	131	Completed	by km

[1] Номера страниц в оригинальном издании

Содержание

СПЕКТРУМ В ШКОЛЕ 9

1. "LISTNAME" 10
2. "REMONT" 10

BETA BASIC 11

ГЛАЗА 1. ВВЕДЕНИЕ 12
Особенности работы с языком BETA-БЕЙСИК 3.0 13
Работа с более ранними версиями БЕТА-БЕЙСИКа 13
Загрузка и выгрузка программ, написанных в БЕТА-БЕЙСИКе 3.0 14
ГЛАВА 2. РЕДАКТИРОВАНИЕ 14
Курсор текущей строки 14
Переключение режимов курсора 15
Управление вводом ключевых слов 15
Ввод ключевых слов БЕТА-БЕЙСИКа 15
Проверка синтаксиса 15
Команда LIST FORMAT 16
Команда CSIZE 16
Команды JOIN <номера строк> и SPLIT 16
ГЛАВА 3. ПРОЦЕДУРЫ 16
Передача параметров в виде ссылки 19
Передача параметров списком 20
Рекурсия 21
ГЛАВА 4. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ 22
ГЛАВА 5. ОБРАБОТКА ДАННЫХ 22
ГЛАВА 6. ГРАФИКА 22
ГЛАВА 7. СЕРВИСНЫЕ И ОТЛАДОЧНЫЕ ВОЗНОЖНОСТИ 23

ЗАЩИТА ПРОГРАММ 24

ЧАСТЬ 1 24
ГЛАВА 1. ИСКЛЮЧЕНИЕ ВОЗМОЖНОСТИ ОСТАНОВКИ ПРОГРАММ. 24
1.1. Обшие рекомендации. 24
1.2 Подпрограмма обработки сообщений с кодами D,H,L. 25
1. 3. Подпрограмма обработки сообщении об ошибке 27
1.4. Метод защиты, основанный на совмещении Бейсика и программы в машинных кодах. 27
1.5. Оригинальный метод защиты, использующийся при загрузке машинных кодов. 29
1,6. Метод защиты, используемый в программе FIST III. 30
ГЛАВА 2. МЕТОДЫ ЗАЩИТЫ ОТ ЛИСТИНГА ИЛИ КАК СДЕЛАТЬ ТЕКСТ ПРОГРАММЫ НЕЧИТАЕМЫМ. 30
2. 1. Общие рекомендации. 31
2.3. Использование управляющих кодов. 34
2.3.1. Введение 34
2.3.3. Управляющие коды ZX SPECTRUM 38

40 ЛУЧШИХ ПРОЦЕДУР 40

РАЗДЕЛ А 40
1. ВВЕДЕНИЕ 40
Общие сведения о БЕЙСИКЕ и машинных кодах 40
2. ВНУТРЕННЯЯ СТРУКТУРА ZX SPECTRUM 41
Карта памяти 42
Дисплейный файл 44
Атрибуты 44
Буфер принтера 45
Область программ на BASIC 45
Цифровой пятибайтныи формат 48
Область переменных 48
Подпрограммы ПЗУ 48
3. МАШИННЫЙ ЯЗЫК Z80 49
Регистры Z80A 49
О системе команд процессора Z80 52
РАЗДЕЛ В 53
4. ВВЕДЕНИЕ 53
Загрузчик машинного кода 53
5. ПОДПРОГРАММЫ СДВИГА. 56
5.1 Сдвиг атрибутов влево. 56

5.2 Сдвиг атрибутов вправо. ... *57*
5.3 Сдвиг атрибутов вверх. ... *57*
5.4 Сдвиг атрибутов вниз ... *58*
5.5 Сдвиг влево на один символ. ... *59*
5.6 Сдвиг вправо на один символ. ... *59*
5.7 Сдвиг вверх на один символ ... *60*

МАСТЕРФАЙЛ 09 ПОЛНАЯ РУСИФИКАЦИЯ. ... **62**

ВСТУПЛЕНИЕ. ... 62
1. Использование символов UDG-графики ... 62
2. Использование дополнительного символьного набора. ... 64
.1. Перевод программы на русский язык. ... 68

СОВЕТЫ ЭКСПЕРТОВ ... **70**

STALINGRAD ... 70
Введение. ... *70*
Управление игрой ... *71*
Некоторые детали ... *73*
Уровни сложности игры ... *73*
Выгрузка файлов на ленту ... *73*
Замечания ... *74*
FLIGHT SIMULATION ... 74
Аспекты полета ... *74*
Приборная панель ... *75*
Управление самолетом ... *76*
Навигация ... *77*
STAR RAIDERS ... 77
Враги ... *77*
Запуск игры ... *78*
Экран ... *78*
Управление ... *79*
Энергия ... *79*
THE TRAIN ... 80
Краткий сюжет ... *80*
Настройка программы ... *80*
Экран ... *81*
Управление ... *81*
Мост (BRIDGE) ... *82*
Станция (STATION) ... *82*
Развилка или перекресток (SWITCH) ... *82*
DEATH WISH 3 ... 83

FORUM ... **85**

Проблемы ELITE ... 85
Вопросы совместимости. ... 91

СПЕКТРУМ В ШКОЛЕ ... **98**

К УРОКУ ИСТОРИИ ... 98

POKES ... **100**

NETHER EARTH ... *100*
HEAD OVER HEELS ... *100*
INTO THE EAGLE'S NEST ... *101*
URIDIUM ... *101*
AMAUROTE ... *102*
GAUNTLET ... *102*

BETA BASIC ... **103**

Раздел 2. Команды ... 103
1. ALTER <атрибута> TO атрибуты ... *103*
2. ALTER <ссылка> TO ссылка ... *103*
3. AUTO <номер строки> <,шаг> ... *105*
4. BREAK ... *105*
5. CLEAR число ... *105*
6. CLOCK число или строка ... *106*

7. CLS <номер окна> *108*
12. DEF KEY односимвольная строка; строка *112*
13. DEF PROC имя процедуры <параметрх, REF параметр> *113*
14. DELETE <номер строки> то <номер строки>. *113*
15. DELETE имя массива <предели> или DELETE строка <цредеяы> *114*

ЗАЩИТА ПРОГРАММ **115**

2. 3.4. Управляющие коды, используемые для защиты от просмотра. *115*
2. 3. 5. Практическое использование управляющих кодов для защиты. *123*
ГЛАВА 3. МЕТОДЫ ЗАЩИТЫ ОТ MERGE. 127

40 ЛУЧШИХ ПРОЦЕДУР **130**

5.8 Сдвиг вниз на один символ. *130*
5.9 Сдвиг влево на один пиксел. *131*
5.10 Сдвиг вправо на один пиксел. *132*
5. 11 спвиг вверх на один пиксел. *132*
5. 12 Сдвиг вниз на один пиксел. *134*
6. ДИСПЛЕЙНЫЕ ПРОГРАММЫ 135
6.1 Слияние картинок *135*
6.2. Инвертирование экрана. *136*
6.3 Инвертирование символа вертикально. *137*
6.4 инвертирование символа горизонтально. *137*
6.5 Вращение символа по часовой стрелке. *138*
6.6 Изменение атрибута. *140*
6.7 Смена атрибута. *140*
6.8 Закрашивание контура. *141*
6. 9 построение шаблонов. *146*
6.10 Увеличение экрана и копирование. *149*

МАСТЕРФАЙЛ-09 ПОЛНАЯ РУСИФИКАЦИЯ **154**

ЗАКЛЮЧЕНИЕ. 163

СОВЕТЫ ЭКСПЕРТОВ **167**

PROFESSIOHAL TENNIS 167
1. CONTROLES *167*
2. EQUIPAMIENTO *167*
3. ENTRENAMIENTO *167*
4. CARACTERISTICAS *168*
0. TORNEO *168*
SNOOKER 168
Правила игры. *169*
Настройка программы. *171*
Структура экрана. *171*
Начало игры. *171*
QUAZATRON 172
CAPTAIN FIZZ 178
Экран программы. *178*
Полезные советы. *180*

FORUM **181**

Версия 1. ..... *181*
Версия 2. ..... *181*
Версия 3 ..... *181*
Перелеты к двойным звездам и невидимым звездам. *183*
ТАЙНЫЕ ВОЗМОЖНОСТИ КОМПЬЮТЕРА. 183
Советы и секреты. *185*

СПЕКТРУМ В ШКОЛЕ **191**

BETA BASIC **194**

16. DO *194*
17. DPOKE адрес, число *196*
18. EDIT <номер строки> *196*
19. EDIT строковая переменная *197*
20. ELSE <оператор> *197*

21. END PROC 198
22. EXIT IF <условие> 198
23. FILL x,y..... 198
24. GET числовая переменная 199
26. JOIN <номер строки> 201
28. KEYIN строковая переменная..... 204
29. KEYWORDS число. 204
30. LET переменная = число <. переменная - число... 205
31. LIST <номер строки> то 205
33. LIST DEF KEY..... 206

ЗАЩИТА ПРОГРАММ 207

Глава 4. прочие приемы защиты.....207
4.1 Запуск программ в кодах. 207
4.2 Защита, основанная на использовании вариаций,числа PI внесто цифровых констант..... 208
4.3 Ключевые слова Бейсика в "хэдере". 209
4.4 йерпающии заголовок. 210

ТЕХНИКА ВЗЛОМА КОМПЬЮТЕРНЫХ ПРОГРАММ ZX-SPECTRUM..... 214

Глава 1. Введение 214
Общие рекомендации. 214
Структура фирменной программы. 215
Глава 2. Блокировка автозапуска. 219
Введение..... 219

40 ЛУЧШИХ ПРОЦЕДУР..... 222

7. ПРОЦЕДУРЫ ОБРАБОТКИ ПРОГРАММ. 222
7.1 Удаление блока программы. 222
7. 2 Обмен токена. 223
7.3 Удаление REM-строк. 224
7. 4 Создание REM-строк. 226
7. 5 Сжатие программы 228
7. 6 Загрузка машинного кода в DATA-строку..... 230
8. Инструментальные программы 233
8. 1 Определение размера свободной памяти. 233
8. 2 Определение длины БЕЙСИК-программы. 233
8.3 Определение адреса БЕЙСИК-строки..... 234
8.5 Копирование данных в памяти. 234

ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ 236

КАНАЛЫ И ПОТОКИ 236

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД 240

Обработка ошибок в БЕЙСИКе 240
Предотвращение остановки БЕЙСИК-программ.....241

FORUM 248

Адвентюрные игры. 249
Heavy on the Magic. 249
Полезные советы. 252
Проблемы совместимости. 253
Доработка "Pentagon-48K' для обеспечения совместимости с Sinclair "ZX-Spectrum" 255
Проблемы "ELITE". 257
47-ая Галактика: тупик или дорога к новым мирам? 259

СОВЕТЫ ЭКСПЕРТОВ..... 263

ACADEMY (TAU CETI II).....263
Работа с меню. 263
Просмотр/выбор клавиш..... 264
Прием нового кадета. 264
Меню работы с лентой. 265
Рапорт о выполнении уровня..... 265
Выбор скиммера..... 265
Проектирование скиммеров..... 265
Выбор миссии. 266

Уровень I. ... *266*
Уровень II. ... *267*
Уровень III. ... *268*
Уровень IV. ... *269*
Уровень V. ... *270*
Полезные советы по сборке скиммеров. ... *270*
Выполнение миссии. ... *272*

SHERLOCK ... **274**

Понедельник, 8:00 ... *274*
Вторник. ... *276*
Среда. ... *277*

НАШ КОНКУРС ... **279**

ZX - РЕВЮ

N1,N2 1992

107241, Москва, Б-241, а/я 37, "ИНФОРКОМ"

Дорогие друзья!

Мы встречаем второй год существования "ZX-РЕВЮ". Позвольте поблагодарить наших верных читателей прошлого года, подписавшихся и на этот. Благодарим также и тех, кто в это трудное время поверил в нас, поверил в то, что мы будем тать и работать и впервые подписался на вале издание.

Мы, конечно, сохраним тот дух и стиль, которые были достигнуты в прошлом году, но кое-какие незначительные изменения все-таки внесем.

Во-первых, уже в этом номере вы увидите больше крупных учебных материалов, этим мы несколько парируем ту раздробленность, которая была в прошлом году. Но в прошлом году мы готовили материал с "колес", а к этому году подошли с определенным портфелем и теперь можем давать более крупные вещи. В связи с этим будет как бы меньше число разделов в одном номере, но каждый из них будет крупнее. По всей видимости, это вызовет также необходимость полностью перейти на практику выпуска двойных номеров. Это, кстати, несколько поможет нам чуть-чуть снизить потери от резко возросших почтовых тарифов.

Поскольку в проекте намечены крупные материалы, мы будем их давать блоками, содержащими четное количество страниц. Это позволит Вам при желании без проблем расшивать выпуски и комплектовать свои подшивки по отдельным работам. Об этом просили многие читатели уже давно, но только сейчас появилась такая физическая возможность.

В этой году несколько большие упор будет сделан на основы программирования в машинных кедах и аа АССЕМБЛЕР е.

Скорее всего, несколько уиень шггся объем материалов технического характера, аппаратурных вопросов. ВЫ ионянлрте. что это связано прежде всего с тем. что кы не занимаемся аппаратный обеспечением к кояьзовзлнсь только тем. что пак лапали партнеры.

А и обшем "ZX-РЕЗ»" останется тек же. кзкия и было.

ны ЦО-ПРРХНСНУ. как и год назад, гарантируем всем читателям, подписавшимся на "РКНЮ" возврат стойкости подписки, если кто-либо разочаруется в содержании это го номера и вернет нам его нлло-хениын платежом, оценив в стои-

мость своих затрат и уведомив нас об этом письмом.

Многих волнует вопрос, как кы выйдем из того положения, в которой оказалась печать в насита-бах всея страны. За ведь знаете, что вэдорохашая в ПРОШЛОМ году в ю РЗЭ бумага сейчас поднялась еще в 5 раз. невыносимыми стали почтовые тарифы, достаточно сказать, что одна бандероль в Прибалтику теперь может стоить до 30 рублей. Люди волнуются, предлагают помошь. Многие пишут, что если нам будет совсем невмоготу, то они лучше доплатят, чем кы закроем наше дело. Большое спасибо. уважаемые читатели за Валу заботу, но пока мы держимся. Наш принцип такой: •Когда дует сильный ветер перемен, то надо строить не забор, а мельницу,

Не успев начать подписной год. вентральная пресса громко заголосила о невозможности выполнить взятые перед подписчиками обязательства и дооилась дотаций от правительства, хотя голосить не перестала, нам такой подход кажется не очень красивым, кы же работаем над другими проектами и сейчас для нас пока нет веских оснований предполагать, что нам придется просить Бас делать какие-либо доплаты, если кы нормально развернен маркетинг программного обеспечения для 1ВП. хотя, конечно, в этом

вопросе НУ очень и очень расчитываен-на валу поддержку.

В N11-12 за прошлый год кы подробно расписали, как мы будем строить нашу с вами деятельность, предполагая начислять тем из Вас. кто покопает продвижение наших продуктов по вх от объема продаж, проведенных п& результатам Вашего исследований местного рынка.

Тан же кы объявили о программном комплексе "АКГРЛИ" (полный к ур с д ir гл кис ко го я эыка в 2 2 - ж программах) и предложили потении -алыши дистрибуторам приобретать для представительства так называемый "ЗЕЛЕНЫЙ ПАКЕТ'. В который входит образец программного средства, рсклакио-инфсркаоионные материалы, бланки договоров. инструкции и т. п.

сейчас, когда пннутся эти строки, процесс уже пошел и по гноим темпам превосходит нашк са-кые смелые ожидания. Работа с дистрибуторами пошла. Если темп бу-дот нарастать так. как сегодня. мы к осени этого года введен вну-

треннее специализированное издание, распространяювееся бесплатно -среди своих-, содержащее анализ работы, обмен передовым опытом и описания наиболее сложных игровых программ для IBH-совместимых машин в качестве развлекательной части, предполагаем, что записывать понравившиеся им программы ванн дистрибуторы будут У вас бесплатно, а коллекция У нас -немалая, одним еловой, определяется наша маркетинговая политика на многие ближайшие годы.

сегодня кы представляем на последней странице "ZX-РЕВЮ" не обучающую программу, а инфорнаои-ОННО-ШШСКСВУЮ систему •РЕГИСТРА-ТУРА". Она выбрана для представления. Поскольку обладает наиболее универсальным характером возможного применения. наверное очень трудно представить организацию, в которой был бы компьютер, чтобы она не нуждалась в подобной системе (разве что аналогичная уже есть), система проходит опытную эксплуатацию ухе более года в разных организациях в качестве системы учета кадров, системы ведения штатного расписания, сне теки учета клиентов в налои предприятии, системы учета обратившихся а медицинское учреждение, системы учета движения документов в делопроизводстве райисполкома, системы учета нгж~ давшихся ветеранов войны и труда, системы учета учащихся на факультетах высшего учебного заведения. Ны внедрили ее даже на одной машиностроительном предприятии для учета заготовок, это то. что ухе есть на практике, а теоретически можно учитывать все и всюду.

имеющийся опыт эксплуатации возвелнет однозначно сказать, что среди систем с аналогичными возможностями нашу выделяет необычайная простота в освоения и эксплуатации, она сделала для тех. кто впервые в своей жизни увидел компьютер, в этом мы видим свое главное достижение. имея опыт преподавательской работы и методологию подготовки обгчаюиих ПРОГРЕЛИ, мы и деловые системы делаем с расчетом на то. чтобы осваивать их можно было без головной боли «без необходимости изучать объемную сопроводительную докукенташвх

Далее, в последующих выпусках ZX-РЕВС кы будем чередовать представление обучающих и деловых систем.

Ждем вашего.участия в дистрибуторской сети.

в заключение кы должны извиниться за возможную нерегулярность выпусков. Хоть мы и с оптимизмом смотрин в будушее, нам все-таки приходится где возможно экономить и. к сожалению. кы вынуждены печатать тираж не тогда, когда надо, а тогда, когда удается купить бумагу по бодее-ненее сносным ценам, т. е. мы. конечно, не упускаем возможности уменьшить свои издержки. В то же время, нерегулярность не означает хроническое отставание, иногда ны. наверное, будем и забегать вперед.

Спектрум в школе

Сегодня раздел для начинающих представляют две БЕЙСИК-ПРОГРЗК мы с комментариями, подготовлен ные Черкасским в. А (Р.БОРИСОВ).

Программы не сложны, но каждая из них содержит ошибки. Попробуй те их отыскать

1. "LISTNAME"

эта прогаммл записывается г начале кассета и при работе ВЫБО -дит названия программ и их место положение оо счетчику.

```
10 DIM A$(50. 3): DIM B»(50, 10):
DIM C*(50.и)
KO BRIGHT l: CLS: PRINT ttO;INK I. FAPER b; AT 0. i; "BORISOV * ALCOSOFT * 1991 ": |
      PRINT AT 8.6; "l.Edlt cbecK"; I AT 9.8: V.LooKine"; I AT 10. e; "3.save to the l|
      tape"
30 IF IHiEY*-";1" THE» GO TO 14C !<Ю IP INKEV* "1- THEN GO TO TC
50 IF IHXEV*-"2" THEN GO TO 100
60 GO TO 30
7Q CLS: PRINT " 'O'-END" 80 IHPU1 "Number "• i:
PRINT " -;i;TABCO) IMPUT "Count-"; a$Ш' -Яаше-";B*(1)' "DATA-";C*U) :  GO SUB 150 90 GO
      TO SO
loo CLS: PRINT - Count Каше
Data ": кок j. j то 50 iiO BEEP .07.35: IF B»{1) = " "
ТПЕК PAUSE 0: GO TO 20 120 GO SUB 150 150 NEXT 1 140 CLS: PRINT AT Ю.10;
"Save proerajns":
SAVE "L1STSAME- ЫНЕ 100:
CLS: PRINT AT 10, 10;
"VERIFY programs":
VERIFY "LJSTNAHE";
GO то го
150 PRIST a$(i); "..._.._.. "; B*U); •_______";C*(i): 8ETURH
```

Пояснения к программе: Ю Задаем строковые массивы для переменных:

a$ - показаний счетчкча; B* - названия программ; C* - год выпуска программы.

20 Выводим МСНЮ.

30-60 Выбор режима работы.

В зависимости от нажатой клавиши травление передается на разные строки программы.

fio переход на строку зо для организации отдания нажатия клав1ши.

70 Стирает экран, выводит условие вызола из режима.

во Ожидается ввод порядкового номера, вывод его на экран и последовательно ввод показания счетчика, названия программы, года выпуска, заканчивается обращением к подпрограмме вывода этих переменных на экран.

90 организуется ввод каталога.

100 Инициализация экрана, вывод заглавия, начало цикла вывода данных.

11С подача звукового сигнала, проверка на "пустое название", если "да", то пауза до нажатия любой клавиши и возврат в меню.

120 Обрашение к подпрограмме вывода переменных

130 конек цикла FOR... NEXT, пауза до нажатия клавиши и возврат в меню.

140 очистка экрана, запись на лент? с последующим автостартон программы с юо строки очистка экрана, проверка записанной программы, возврат в меню

150 подпрограмма вывода переменних

Ошибки нужно добавить строку

145 If 1^0 THEM GO TO ЮС

и в строке во добавить после ко манды INPUT "Huraber=": j

: GO SUB 141

это даст нормальный выход при программирования каталога и выве дет название. KOTGPO* пояьзова тель будет редактировать

Кроне того, в 1Ю строке сравне ние нужно проводить не с " '. ас т. У. переменная в»(1) имеет фиксированную длину в

10 СИМВОЛОВ ЭТО ИМИ 11"ОГР,1ММЫ

2. "REMONT"

эта программа может СЛУЖИТЬ пособием по ремонту различной техники. Причем,

когдз у машины варианты кончатся, то она спросит у пользователя, что неисправно, и в следующий раз она задаст и этот вопрос. Таким образом, постепенно количество вариантов увеличится.

```
5 GO TO EO
10 LOAD "CODE
го DIM a$i5oo.зг>
30 LET II:
LET a$(П-" напряжение батаре &к"
to CLS: POKE J3607,249: PK1HT"Я дунаю. стоит проверить "; a$ U):  GO SUB 140 50 IF UP-O
      THEN PRJUT FLASH 1';
" НУЖНО ремонтировать'.":
PAUSE 0: GO TO 90 50 LET l=l+l:IF a$(l)=- •
THEN GOTO 80 70 GO TO *O
60 INPUT "Не хватает вариантов "' " Как ви думаете, что erne НУЖНО проверить?" 'a$ a) 90
      POKE 23607, БО: CLS: PRINT AT Zl.R
"Press any Kev '": PRINT #0: IKX 1; PAPER 6;AT 0.1; " BORISOV * ALCOSOFT * 1991": POKE
      23607,249: PRINT AT 5,3:
"1. - Работа с программой": AT Ч, 3;
"2. Запись на магнитофон"; AT 7. з;
"3. - проверка записи" 100 IF INKEY*-"!" THEH GOTO 30 ПО IF IHEEY*;"2" THEN SAVE
•REMOHT" LIHE 10 130 IK INKEV* "3" THEN VERIFY
"REMOST" 130 GOTO 100
140 PRINT ," в норме?"'" Да/Нет7" 150 IF PEEK 23560M00 THEN LET
UP'I:        RETURK
160 IF PEEE 33560-110 THEN LET 1
UP-0:        RETURN !
170 GOTO 150 !
```

Описание I

ь служит для того, чтобы пос ЛР команды RUN программа нормаль-! но стартовала, а не- производила j ЗЭГРУЗКУ по стгоке iO. I

io загружает коды энакогенерз [тора РУСКОГО алфавита.

20 Объявляет строкОВЫЙ нас I

СИР a$

30 Объявляем переменную , и элемент массива a$Ш.

'K Считаем экран переключаемся на русский шрифт, вынодим! вопрос и обращаемся к подлрограмме управления

50 ЕСЛИ неисправность определе на цравильно, т<: выдаем сообше-ние

ьс Прибавляем і к і. вере* ш? ррменнуто АМз? ;• провернем р' Н=і

II УС ТОТУ

70 организовываем переход для сяедуююего восроса.

80 Хден ВОПРОС, КОТОРЫЙ вво дит пользователь. 9О вывод меню.

500 ОПРОС клавиши "1" Если на жата, то работа с программой.

но То же клавиши "2". Если нажата - выгруика программы на ленту.

120 То же клавиши "3". Если на жата - проверка программы.

130 Организовывает ожидание INKEV*.

?«о Начало подпрограммы определения ответа.

150 опрашивается клавиша rD".

ibo То же - клавиша *Н".

1?о организация ожидания INTEY». знакогенератор наводятся в памяти по адресам 64000. . . б47бв

Ошибки,

1. в строку 150 надо дописать POKE E356O, 0 для того. чтобы программа не проскакивала IHEET*.

г. в строке 60 сравнение должно быть с пустой строкой длиной з? символа

BETA BASIC

БЕТА БЕЙСИК 3.0 - дальнейшее развитие серии, начатое программами БЕТА- БЕЙСИК

1.0 и 1.8. Как и последние, он полностью совместим со стандарты»!, защитам в ПЗУ Вашего компьютера БЕЙСИКОМ, но обладает значительно большими возможностями, определенной гибкостью и имеет значительное число новых операторов и функций.

БЕТА БЕЙСИК откроет перед вами гиганте кие перс пектиеы. Во-пер-вых. с его ионошью Вы можете писать большие программа. Если вы не пробовали на обычном Бейсике писать программы более 10 - 15 К. то знайте, чп> "проклятьем" обычного БЕЙСИКа является отсутствие структурного программирования, наступает момент, когда программа не укладывается, как цельное произведение в Вашем мозгу и тогда любые изменения, производимые в одном месте программу, влекут за собой помехи, появляющиеся в другом ;*есте. Как бы ни был прост бейсик а освоении, но 'а отладке большие программы. написанные на БЕЙСИКе, могут выходить за все разумные рамки но трудоемкости.

второе преимущество бета-беи-СИКа отличные средства редакти-ро вания. "JTOT момент. ;1адо ска -члть. Jхиляесова пята сганпарт кого БЕЙСИКа "Спектрума". конечно. строчный редактор со слож-ной системой вызова строки на обработку - это не дело. Только на этом программист теряет ноновину своего драгоценного времени. в то же время. полноэкранные редакторы, какие мы видем в ней -сике на хорошо продвинутых на шинах типа нзх ил.ч. •••кажем, ьвн (GW.BASIO - это тоже не помарок. Перемешанные на одном экране редактируемые и нередактируемые строки приводят к многочис ленным ошибкам, да и занимают такие редакторы много места.

Тот подход. который развит в бета бейсик i.Q. наверно* наибо лее практичный и эргономичный.

Третье преимущество !>ЕТА БЕИ сика з ЙРО .жономном расходовании памяти. Да. конечно, сам он занимает до ^ОК. -'г> есть почти половину того, что имеет ваш юн аьютер. Но. УСЯИ аы. лрограмми рун. будете ХОРОШО использовать его возможности. то затратите 4 б К на основные .кроаедуру. далее, используя их. полете ч каждый килобайт укладывать vo. if ч тс' ч« ^ычн",н ;:касик* ---«ло

бы Вам пяти-семи К. Таким образом, в ДОСТУПНЫХ вам го килобайтах, вы сможете разместить примерно ЮЕ данных плюс ч-5 к процедур плюс пропрамну. эквивалентную обычнам 4ок стандартного БЕЙСИКа. конечно. это цифры условные. но условность ИХ Э ТОН.

что это далеко не предел, поработав несколько недель. Вы сможете еще сильнее повысить емкость программ, это просто данные нашего собственного практического опыта в "ИНФОРКОМе".

данный документ является наиим авторизованным переводом Фкрмен ной ИНСТРУКЦИИ. Авторизованность не означает, что мы что-то сократили, ни чуть более содробно ЗРО-комментировали наиболее узкие места и заменили (там. где это можно) те примеры, которые были привязаны к микродрзиву на более близкие для вашего читателя.

ПУСТЬ вас не смукает объем документа. Нет никакой необходимости читать все ст начала и до кон-па, вполне достаточно, если Вы прочтете введение, раздел по редактированию программ и, в дальнейшем, будете обращаться к документу только при необходимости использовать тот или иной оператор или ту или иную, ранее неиз вестнтю Вам Функцию. Особо же рекомендуем вам обратить внина-:ше на "е :'ре имущества, которые предоставляет структурное ПРОграммироианис благодаря использованию процедур.

ГЛАЗА 1. ВВЕДЕНИЕ

Загрузка язика с ленты выполняется обычным обРыЗом:

LOAD -" или LOAD "Beta Basic-

по этой команде загружаются ТРИ первых ЧЕйсик-строки - строка о, строка 1 и строка 2. со строки г осуществляется автостэрт программы, в результате которого загружается остальная (основная) часть БЕТА-БЕЙСИКа. представляющая из себя блок машинных кодов длиной 18 к. системная переменная RAMTOP понижается до значения "1707о, предохраняя от повреждения машинный код. размещаемый в верхней -ысти памяти.

после загрузки программы на жране должно появиться исходное :истемное

сообщение. при JTOH ГТРОКИ 1 и ? программы удаляются .1 ;-сТсн>тся только нулевая строка.

^нинаяие! ;сли мы . «аписали программу :юд управлением БЕТА SEHCmsa л желаете передать ее тзоим .шакомым для эксплуатации, •'•". -'^жйлувста ;iepen ВЫГРУЗКОЙ ..F-ign РОграммы и машиннокодового .к.к,- snecHve следующие изменения "1,1:'1нн»окодовую часть ,1;ЕТА-БЕИ- :;;Ko

•-;-' Wt.'b,'-j: TOKK 5ЧГ'19. О:

РОКЕ б^гго, о: РОКБ &здб1,го1 этим вы обеспечите то. что ваша программа бгдет запускаться, но не сможет быть изменена.

Особенности работы с языком ВЕТА-БЕЙСИК 3.0

В строке о содержатся новые определения функдий БЕТА-БЕИСнка 3.0. эта строка всегда присутствует в программе, хотя и не всегда изображается на экране. характерным свидетельством того, что БЕТА-БЕЙСИК загружен и готов к работе является изображение указателя текущей строки в инвертированном виде. Несколько увеличена и продолжительность ЗВУКОВОГО сигнала, подаваемого при нажатии клавиш. Если он вам не нужен, дайте команду POKE 23609.0 для отключения, теперь можете загружать любую БЕЙСИК-программу [ДОСТУПНОГО размера) и она будет работать нормально, за исключением ДВУХ особенностей: J. Загрузку программы надо выполнять не командой LOAD, а командой MERGE. иначе погибнет нулевая строка, конечно, если программа была написана под стандартным БЕЙСИКОМ, ей это все равно, :ю если под БЕТА БЕЙСИКОМ. строку о уничтожать не надо, г. ЕСЛИ вам надо, чтобы вместо ключевых слов БИТА-БЕЖГИЯа на экране изображались символы блочной графики, надо дать команду KEYWORDS о (см. соответствующую команду).

Кше одной особенностью является то, что несколько изменилось изооражение программных строк. получаемое на экране по команде LIST, если длина строки больше ШИРИНЫ экрана, в этом случае в первых четырех позициях печатаются только номера стрск.

Несколько уменьшено рэзрушительное действие команды HEW. Она удаляет из памяти имеющуюся тан БЕЙСИК программу, но оставляет строку о. Если же Ба хотите удалить из памяти и сам БЕТА-БЕЙСИК, то остается только сделать RESET, если У вас есть такая кнопка . или выключить питание или переинипиадиэировать машину командой RANDOMIZE US* О.

Несколько увеличена скорость выполнения программ но сравнению со стандартным 5ЕИСИКом. особенно длинных.

Циклы FOR-NEXT работают с од-чой и той же скоростью, независимо от местоположения в программе. что является отличием от стан ДЗРТНОГО БЕисика. наивысшая ско РОСТЬ достигается, когда начало -гикла, t>ro длина и конец выражены целыми числами ;в том числе и от рииательными) Размером до 6553Э. кие циклы работают в 5 Раз быс-":•(-!-'. t'o --равнр-нию с циклами.

разношенными до юо-и строки в стандартном БЕисихе и в IT раз быстрее, по сравнению с размешенными до 5ОО-ой строки в стандартном БЕИСНКе.

Быстрее работают и GO то и GO SUB, особенно когда строка назначения отстоит достаточно далеко. Быстрее работает и RETURN. возврат к последней строке выполняется столь же быстро, как и к первой, в отличие от стандартного БЕЙСИКа.

В какой-то степени скорость удалось повысить за счет того. что интерпретатор запоминает адреса, а не номера строк, с другой стороны, это накладывает определенные ограничения на редактирование, так. если вы, например, остановите программу во время исполнения цикла. введете внутрь этого цикла дополнительную строку, а потом запустите программу дальше - CONTINUE. то CONTINUE может не сработать. поскольку адрес возврата изменится, хотя номер строки возврата не изменялся.

Работа с более ранними версиями БЕТА-БЕЙСИКа

Если вы загрузите программу, написанную в более ранней версии БЕТА-БЕЙСИК 1.0 ИЛИ БЕТА-БЕЙСИК 1.8. то вместе с ней загрузится и нулевая строка, содержащая

определения функций языка. Если теперь Вы хотите работать с этой программой под управлением версии 3. 0. то вам надо заменить старую версию нулевой строки на НОВУЮ.

1. Загрузить БЕТА-БЕЙСИК 3.0.

2. загрузить программу, выполненную в версии i.o или i.e.

3. ВЫПОЛНИТЬ "MERGE" ДЛЯ БЕТА-БЕЙСИКа 3-0. чтобы ввести

КОПИЮ строки О.

%. удалить строки 1 и 2.

5. Выгрузить полученную ПРОГРЗИНУ.

Есть два момента несовместимости версии 3.0 с ранними версиями. Во-первых, переменная иод именем •line", создававшаяся one раторани он ЕйВОй и TRACE, теперь называется "lino", чтобы исключить путаницу с ключевым словом стандартного БЕЙСИКА LINE.

ВО-вторых, в новой версии имена процедур не МОГУТ иметь внутренних пробелов.

Преобразовать Вашу программу из версии, написанной на более ранней версии языка, вы мохете воспользовавшись командой ALTER.

для тех. кто знает версии 1.0. 1.в. желательно перечитать работу

С командами ОН. REMUH. ROLL, SCROLL. CLOCK. ON ERROR И TRACE,

ПОСКОЛЬКУ они изменены, несколь ко изменена и команда GET (число), значительно расширены возможности продедур.

Загрузка и выгрузка программ, написанных в БЕТА-БЕЙСИКе 3.0

После того, как вы написали программу, в которую входят некоторые новые ключевые слова, вы можете выгрузить ее на ленту обычным порядком. Если теперь вам когда-либо понадобится вновь загрузить ее. сначала загрузите БЕТА-БЕЙСИК 3.о, если ок еще не загружен, теперь загрузите (LOAD) свою программу. поскольку строка о была выгружена вместе с программой, то теперь нет необходимости использовать MERGE.

С другой стороны, вы можете воспользоваться строками 1 и г загрузчика для того, чтобы сохранить на ленте свою программу вместе с нашиннокодовой частью БЕТА-БЕЙСИКа. в этом случае ваша программа и БЕТА-БЕЙСИК будут загружаться автоматически.

Если вы загрузите программу, написанную на БЕТА-Бкисике. в компьютер в то время, как в нем не присутствует машинный код БЕТА-БЕНСИКа. новые команды появятся в виде одиночных символов. а после команды RUN вы получите сообщение об ошибке "Nonsense in Basic", в этом случае вам следует ис поль зовать:

MERGE "Beta Basic": GO то г. чтобы загрузить загрузчик БЕТА-БЕЙСИКА, а из него ио автостарту посредством со то г загрузится и машиннокодовая часть.

ГЛАВА 2. РЕДАКТИРОВАНИЕ

Список используемых ключевых слов: EDIT. KEYWORDS. LIST, FORHAT. CSI2E. JOIH, SPLIT.

БЕТА-БЕЙСИК позволяет делать ввод и редактирование программ намного более простым, чем то. к чему вы привыкли, работая со стандартным встроенным БЕЙСИКОМ "Спектрума". Если вы наберете или прильете (MERGE) некуп БЕЙСИК-програмиу. то сможете поэкспери-ментровать с новыми возможностями. Поскольку все возможности стандартного БЕЙСИКа сохранены. Вы вряд ли будете испытывать при этом какие-либо трудности. Ниже мы рассмотрим, как действуют те команды, которые добавляет ис-пользование третьей версии БЕТА-БЕЙСИКа при редактировании. Более подробно мы на ник остановимся еще раз а основной части руководства.

Курсор текущей строки

Первое, что вы заметили, загрузив БЕТА-БЕЙСИК. - это курсор текушей строки, изображающийся инверсным цветом на экране. Его можно перемещать с помопью курсорных клавиш вверх и вниз и выполняется это намного быстрее. чем в стандартном БЕЙСИК?, поскольку при этом не листается со

держиное всего экрана. (Могут быть редкие случаи, когда положе-ние курсора не вполне соответствует изображению текста программы на экране, в этом случае просто нажмите ЕНТЕК для оерестрое-ния экрана).

КОМАНДА EDIT <нонер строки> Вы можете вызвать на редактирование даже ТУ строку, которой в данный номент нет на экране, теперь нет необходимости подгонять к ней КУРСОР, достаточно нажать "О" и набрать НУЖНЫЙ номер строки, чтобы она появилась в нижней части экрана.

в редактируемой строке вы теперь можете перемешать курсор не только влево или вправо, но и вверх и вниз, это опять же выполняется курсорными клавишами. при попытке поднять курсор выше верхней строки или опустить ниже нижней, он автоматически встанет в конце строки, самый быстрый путь добавить операторы в коней вашей строки - это:

- нажать клавишу "О";
- ввести номер строки;
- нажать "курсор вверх" теперь он окажется в конце строки.

Переключение режимов курсора

Теперь вы ножете в нужный но мент легко переключить курсор рк~ на курсор "L" или "С". Это бывает полезным в тех случаях, когда Вы хотите набрать по буквам имя процедуры или если вы не желаете вводить ключевые слова как токе-ны. 1*0 есть одним нажатием клави ши. а хотите набирать их по бук вам. что возможно благодаря ко нанде KEYWORDS сем. нижи», вы полняется переход из режима "К" в режим "L/C" нажатием клавиши "ПРОБЕЛ".

Возможен и обратный переход -из режима "L/C" в режим "К", что выполняется одновременным нажати ем клавиш SYMBOL SHIFT и ENTER. те. кто внимательно читает "ZX-РЕВЮ", знают, что прямым путем в стандартном. БЕИСНКе это невоз можно, для этого мы набирали оператор тнкн. а потом стирали его (сн. 'Маленькие Хитрости" zx РЕВЮ-91. СТР. 5й). это бывает по лезно, если вы работает в режиме с отключенными токенами (KEYWORDS Ч - см. ниже) или если Вы хотите ввести ключевое слово ц строковую переменную, что бывает полезным при работе с командами RKK. AI.TKR ИЛИ KKYIK.

Управление вводом ключевых слов

Команда KEYVORDS позволяет Вам переключать режим ввода ключевых слов. т. е. вводить их одним илж<1 тиен клавиши, как в стандартном БЕЙСИКе. или набирать пилностьи по буквам, как это делается на компьютерах иных систем, Ксть и

Режнм KEYWORDS 3. D которой 8 одной строке можно одновременно применять и тот и ДРУГОЙ подход. это толе может быть полезный, даже если зы корошо знаете стандартную систему набора и Вам нравится набирать слова тина RAHDO-MIZE одним наяатием клавиши, все же ввести оператор IH по буквам несколько проше. причем набор ножет идти как прописными, так и строчными буквами.

Ввод ключевых слов БЕТА-БЕЙСИКа

Есть доз способа ввода ключевых слов БЕТА-БЕВСИКа. Вы можете набирать их по буквам, используя ведущий пробел для того, чтобы отключить курсор "?". если необходимо, или можете использовать "одноклавишный" подход, в последнем случае команды и функции вводятся поразному. Для ввода новой команды сначала перейдите в графический режим, а затем нажимайте соответствующую клавишу. Большинство клавиш в этом случае дают новые ключевые слова.

Для ввода новой функции наберите ?Н. а затем -"*" или "I" - в зависимости от того, что это за функция) . набирать "FK" можно теперь по-разнону. Во-первых обычным порядком, как ключевое слово стандартного БЕИСИЖа, во вторых *до буквам "f" * "n" •+ " " и, в-третьих. - нажав клавишу "V" в графической режиме.

Проверка синтаксиса

БЕТА БЕЙСИК, как и стандартный БЕЙСИК, проверяет правильность того, что Вы вводите в кон-ныотер и точно так же выдает звуковой сигнал BEEP, если устанавливает

наличие ошибки, изменить звуковой сигнал Вы можете, изменяя значения в ячейке памяти 23608 посредством POKE 23608.. . .

Звуковой сигнал удобен, если вы набираете программу, например, из распечатки в журнале и при этом не часто смотрите на экран. Но имейте в виду, что если Вы работаете в режиме набора ключевых слов по буквам, то ошибки в их написании будут восприняты компьютерен как ввод нового имени процедуры, например, если вместо PAPER I вы наберете PAFRE 1. то подучите сооешение об ошибке "H1331П8 DEF PROC' (отсутствует определение процедуры).

Команда LIST FORMAT

Эта команда позволяет Вам улучшить читабельность распечатки вашей программы на экране монитора.

Команда CSIZE

Позволяет вам уменьшить размер символов, доведя их количество в строке до бч- или. наоборот, увеличить их размер для печати заголовкое или в иных, например в демонстрационных пеляя.

Команды JOIN <номера строк> и SPLIT.

Позволяют объединить две строки в одну или, наоборот, разбить строку на две.

Уиравлястнн код "новая строка".

Вы можете ввести этот управлявший код в свою программную строку шш в строку INPUT путем нажатия CAPS SHIFT И ENTFS. В отличие от просто ENTER Вы сможете продолжать ввод той же программной строки на другой экранной строке, смотрите также раздел "Управляющие коды" <CH8* 35).

ГЛАВА 3. ПРОЦЕДУРЫ

Список исиользуекых ключевых слов: DEF PROC, END PSOC, LOCAL, DEFAULT, KEF. READ LIHE. LIST PROC И функция ITEMO.

В этой главе мы кратко рассмотрин вопросы, связаиные-ие применением процедур в БЕТА-Бейсике. Более подробную информацию вы сможете получить далее, прочитав разделы относящиеся к каждому из указанных ключевых слов.

Использование процедур - весьма желанная прием для тех, кто программирует на НЕИСИКе. современные версии БЕВСИКа в той или иной мере используют эту возможность и программисты встречают их с большим энтузиазмом. И, если вы ими не пользуетесь, то наверное стоит пересмотреть свои взгляды, причем вовсе не потому, что это модно или потону, что это все рекомендуют, а просто ^гак удобнее и легче писать и отлаживать программы.

БЕТА-БЕЙСИК 3.0 имеет одну из самык совершенных систем использования процедур среди других языков для домашних персональных компьютеров, именно здесь вы получаете наибольшую гибкость и эффективность программирования.

Основное преимущество исполь-зования процедур состоит в структурном программировании, эта концепция предполагает, что вы можете создать некоторый программный модуль, способный выполнять определенную работу и не оказывать никаких нежедатеяйья побочных эффектов на остальную часть программы, он не должен, например, изменять никакие переменные в программе, кроме тех, которые положено. После того, как этот блок программы Вани написан и тшатель-но отлажен. Вы можете забыть о том, как он работает и из чего состоит. Когда Вы пишете новую программу и он вам нужен, вы просто пришьете его к тексту командой MERGE "", а используете -

Вызвав его по имени, каждая ПРО дедура должна быть достаточно простои, чтобы быть вполне понятной. ЕСЛИ она У Вас получается громоздкой, сложной и непонятной, то стоит подумать о том, чтобы разделить ее на несколько логических частей и выразить каждую из этих частей своей процедурой. Процедура - это часть программы, имеющая свое имя и начинавшаяся с оператора DEF PKOC, после которого задается ее имя и имена тех переменных, с которыми она должна работать, а заканчивающаяся оператором EKD PROC.

Давайте рассмотрим простой. яотя и вполне бесполезный пример:

100 DEF PROC ereet

НО P8INT -Hello"

120 END PROC

попробуйте дать команду кия и вы увидите, что ничего не произойдет. Хоть процедура и часть программы, но работать в таком виде она не будет, здесь мы записали только определение црояе-ДУРН, то есть указали, что она должна делать, вьшолнечше процедуры происходит только после вызова ее по имени, наша процедура имеет имя "ereet". попробуйте добавить строку

ю greet

Теперь, если вы дадите команду кин. в строке Ю будет заптше-на эта процедура и, в соответствии с ее определением, будет на печатано на экране слово Hello.

Все происходит точно так же. как в стандартном БЕЙСИКЕ с функциями. Там ведь тоже DKF FH игнорируется до тех ПОР. пока программа не встретит вызов функции оператором FH,

Вы можете столкнуться с проблемой того, как набрать слово greet в строке Ю. ведь курсор находится в режиме "К", Выше мы об этом упоминали:

вы можете нажать пробел и далее набирать greet по буквам;

вы можете дать команду KEYWORDS 4 и перейти в режим ввода всех ключевых слов по буквам;

- Вы можете использовать ключевое слово PROC и записать стро КУ в виде:

ю PROC ereet

Это совершенно то же самое, что я просто

Ю ereet

использование ключевого слова PROC в данном случае просто дань привычке тех программистов, которые привыкли к такой записи по более ранним версиям БЬТА-БЕЯСИ-Ка. хотя реально в нем больше нет никакой необходимости.

Далее мы будем говорить об использовании имени процедуры, как о вызове процедуры. Имя проиедуры обязательно должно начинаться с буквы- Заканчивается имя нроцедуры пробелок, двоеточием, нажатием KHTKK. операторами REF иди DATA. Остальные символы, как правило, могут быть использованы в имени продедуры. но желательно, чтобы мы привыкли использовать только 6уквы. цифры Н символы подчеркивания - "_". не имеет значения, какие буквы применяются строчные или прописные.

Определение процедуры начинается с DEF PROC и может быть расположено гае угодно з программе. Его вполне можно располагать и до и после первого вызова процедуры. Важно только, чтобы оператор DEF PROC был первым операторе** в строке. определение процедуры может иметь сколько угодно строк и должно заканчиваться оператором END екос.

Если Вы используете с одним

DEF FROC НеСКОЛЬКО END PROC. ТО

Компьютер будет не в состоянии правильно "перепрыгнуть" через определение процедуры во время работы программы, в этой случае вам придется самостоятельно позаботиться, чтобы он смог это сделать или размещать все лишние ESD PROC после оператора STOP. изменим приведенный выше пример; юо DEF екос ereet tiroes i10 FOR n=1 то times 1EO FRINT 'HELLO-130 NEXT Г. \1Q END FROC

В данном случае times - это параметр процедуры, он называется формальным параметром, по-скольку не имеет пока никакого числового значения, а только указывает, как он используется при расчете процедуры. Раз он указан в определении процедуры, то он должен быть задан и при вызове процедуры. теперь, когда вам надо будет исполнить процедуру ereet, вы зададите параметр times и заданное Вани значение называется фактическим параметром - это число, которое фактически будет подставлено на место формального times при расчете процедуры.

Итак, формальные парнетры -это просто имена, а фактические параметру - это числа, выражения или переменные (кроне тех случа-ав. когда используется REF. о чем см. ниже).

Теперь, если вы обратитесь к процедуре по имени, без задания параметра, например 10 ereet . то получите сообщение об ошибке "Variable not found" (переменная не найдена).

Если же Вы обратитесь к процедуре:

10 ereet 5

Вы увидите что слово Hello будет пять раз напечатано на экране, т.к. фактический параметр "5" встал на место формального times.

ны надеемся, что тем. для кого применение процедур в программах является новым делом, пока все понятно, но надо обсудить еще один очень важный момент, который может в больших программах повлечь за собой трудно обнаруживаемые ошибки,

дело в том. что. как мы говорили, процедуры должны исполнять то. что записано в их определении и не должны оказывать нежелательных побочных воздействии на остальную часть программы. Рассмотрим нашу процедуру, она манипулирует с двумя переменными times и п. Лазайте теперь проверим, чему они равны после того, как процедура отработала. дайт? пряные команды:

PRINT times

FRINT П

вы обнаружите, что times - не существует, an- существует и имеет числовое значение.

Почему не существует times ? дело в том, что эта переменная введена нами в строке DEF PROC. что автоматически сделало ее локальной, то есть ош*еделеиной тс-лько внутри процедуры. Когда процедура кончила работать, она удаляется из памяти и ее больше нет. И это хорошо, ведь если У Вас где-то еще в программе используется какая-нибудь переменная times, то ее значение могло бы быть изменено (испорчено), а теперь вам можно об этом не думать.

А если У Вас до вызова процедуры >же была какая-то переменная times, то она тоже будет удалена? нет. в момент вызова она будет запомнена, как глобальная переменная, а в процедуре будет создана локальная tunes. После работы процедуры локальная будет удалена, а глобальная - восстановлена.

Хуже обстоит дело с перемэн-ной п, - она осталась существовать после работы процедуры и является глобальной, ведь она не входила в оператор DEF PROC. ЕВ значение может незаметно дгя Вас быть использовано где-то еще и привести к неприятным ошибкам, для того, чтобы это не происходило, ее можно объявить локальной переменной внутри процедуры в принудительном порядке, для этого служит оператор LOCAL.

Добавьте * нашей программе строку:

105 LOCAL П ,

а теперь добавьте к программе следующие строки, которые позволят вам убедиться, что после работы процедуры переменные times и п остались ненарушены:

10 LET П=1234

го LET times-56?e

зо ereet 10

40 PRINT n. times

Процедура "areef имела скорее

учебное, чем практическое назначение, а вот более полезная процедура:

```
100 DEF FROC bок x. г. width,heleht
130 PLOT X.У: DRAW Width. 0
140 DRAW 0.-heieht: DRAW -Width. 0
150 DRA'W 0.heiflht
160 END FROC
```

Процедура "bos" предназначена для рисования прямоугольника и имеет четыре параметра: x, г -координаты левого верхнего угла, width - желаемая ширина, а hcleht - высота, так. команда

ьох юо. шо. ю, 4О изобразит вблизи центра экрана прямоугольник. вытянутый по вертикали.

теперь предположим, что мы хотим, чтобы У нас был квадрат, а поскольку у него высота равна ширине, то попробуем ее не указы вать, например:

ьох юо. 100. 50

Получим сообшение об ошибке, т.к, процедура имеет четыре форнальвых параметра, а мы им на смену подставили только три фактичес:сих. Это, однако, тоже можно предусмотреть и предотвратить, для чего служит оператор DEFAULT (по английски default - принятый "по умолчанию"), запишем строку:

1EO DEFAULT helght=width

эта строка буквально означает следующее: "Если параметр heieht не задан, то считать, что он равен параметру width'.

НУ, а если вы УЖ совсем ленивы, то можете не указывать и ширину квадрата, введя строку:

110 DEFAULT With - 20

и у вас и ширина и высота будут равны 20 пикселам.

Используя запятые, можно опускать не только последние параметры, но и вообше любые, box . юо.чо. ю

в этом вызове опуиен параметр к. Разумеется, где-то в описании процедуры должен присутствовать оператор

DEFAULT X -... .

Передача параметров в виде ссылки

Итак, мы рассмотрели, как информация передаете я из главной части программы в процедуры с по-мошьо параметров. Фактические параметры заменяют формальные, определенные в операторе DEF PROC.

Но у нас может возникнуть необходимость не только передавать что-то в процедуру, но и. например, получать что-то от нее. можно, конечно, написать процедуру, которая будет что-то рассчитывать, а результат расчета ИРИСВЗ ивать глобальной переменной х. из которой можно этот результат узнать после окончания работы СРО педуры и возврата в главную программу, но это наружит нашу до

говоренность о тон. что процедура должна быть независимым модулем и не должна оказывать, нежелательного виняния на другие участки программы (и на другие процедуры тояе). Л если мы поступим таким образом с переменной х, то теперь должны будем все вреня помнить о тон. что эту букву уже нигде нельзя использовать для иные ае-лей, т. к. можно потерять то, что в этой переменной содержится. Желательно было бы уже при вызове процедуры к указании фактических параметров задать какую-то переменную, в которую надо поместить результат работы процедуры, данная версия БЕЙСИКа. в отличие от многих аналогов, позволяет делать и это.

Обычно. при вызове процедуры в нее передаются параметры Б виде значений (чисел), которые встают на место формальных параметров. Но можно и передать просто имя переменной, без указания ее значения, это называется передачей параметра, как ссылки, и делается добавлением оператора REF перед именем этой переменной в операторе DEF I'ROC. KEF - это сокращение от английского слова reference (ссылка:, таким образом, перемен ная, которую Вы ставите при вызове процедуры в качестве фактического параметра, переименовывается в то имя. перед которым стоит REF. вычисляется в процедуре и воз врашается под первоначальным именем в вызывавшую ирграмму или процедуру. В качестве денонстрации приведен процедуру SWOP, ко торая обменивает иежду собой содержимое двух строковых перемениные.

```
rOC DEF FRQC SWOP REF a$, REF b*
210 LOCAL t*
220 LET t*=a»: LETaJ^b»:
LET b»-t* 230 END FROC
```

А вызывается она. например, так:

```
10 LET X»: 'Ы": LET y*;"eoodbye" 20 SWOP X»,У* 30 PRINT X*. У*
```

Без указания REF в определении функции и a$ и Ь* тоже будут обмениваться своим содержимым в процедуре SWOP, но только именно внутри нее; глобального эффекта не будет, т. к. локальные a$ и ь* при выходе из процедуры будут утрачены, с использованием же REF эти переменные являются как бы временными именами, на которые ссылаются глобальные х* и У*. Поэтому изменения. которым подверглись a$ и Ь* в теле процедуры,

отражаются и на x* и на y*.

Языки программирования для микрокомпьютеров, использующие кон цепции процедур, в большинстве случаев не позволяют использовать массивы в качестве параметров.

БЕТА-БЕЙСИК разрешает это, но прапда. требует, чтобы они передавались только как ссылки, они переименовываете я внес то того. чтсбы прост^ копироваться в гб ласть локальных церемонных процедуры. Этим достигается экономия памяти, ведь компьютеру не надо одновременно хранить один массив два раза. Если же вам на самом деле нужна локальная копия Вашего мае сива для какнк-то временных манипуляций с ним, вы можете внутри процедуры создать параллельный массив, объявив его как LOCAL, а затем скопировать в него содержание вашего исходного массива, воспользовавшись для этого командой БЕТА-БЕЙСИКа COPY, о чем мы еще скажем ниже, когаа будем ее рассматривать вместе с командой JOIN. вот демонстрационный пример, в котором показано, как можно найти сумму элементов массива.

```
300 DEF PROC total REF at),
REF sum
310 LOCAL П
320 LET ЯШП-0
330 FOR П=1 TO LENGTH <l.¨a(>¨)
340 LET sum - sunn + at)
350 ЯEXT П .
360 END PROC
```

за именем массива должны идти скобки, чтобы' интерпретатор отличал имена массивов от обычных пе-рензнных с тем же именем, перед "зша" стоит оператор REF, так что яо окончании работы содержимое "sum" будет передано глобальной переменной. функция LLNGTH (). о которой мы еще будем гонорить. определяет размер массива для того, чтобы процедура могла работать с массивами любой длины.

Теперь зададим сам массив, чтобы Убедиться, что наша процедура работает нормально:

```
100 DIM U10)
HO FOR П=1 TO 10
tao LET t<n>=n
!30 NEXT П
```

и добавим вызов нашей процедуры:

14O total to. answer

150 PRINT answer в итоге получим 55.

Передача параметров списком

Возможны варианты, когда Вам вместо того, чтобы определять конбинаоиг параметров для процедуры, удобнее иметь дело со снискон этик параметров, причем список может быть неопределнной дли ны. чтобы это было возможным, в БЕТА-БЕЙСИКе з. 0 есть специальные средства.

если в операторе DEF PROC ис пользовать оператор DATA вместо обычного перечисления формальных параметров, то соответствующий ему оператор READ примет список фактических парметров. стоящих в вызове Вашей процедуры, чтобы эта возможность была по настоите-ну удоьной, необходимо, чтобы можно было определить я процедуре есть ли еще параметры в списке, которые она не приняла. Для этого существует функция ITEM t). Она возвраяает о, если список исчерпан полностью, 1 если в списке есть непереданные параметры и следующий параметр - число, г если следующий параметр - строковая переменная,

Приведен пример, КОТОРЫЙ дока зывает, как можно организовать процедуру, которая просуммирует несколько заданных Вани чисел. II 100 DEF PROC Sum DATA It 110 3-0

120 DO UNTIL 1ТИШ-0 I 30 RF.AD a

140 з-s+a

150 LOOP

160 FR1HT 3UTO

170 END PROC

примененные в строках 120. iso операторы DO UNTIL и Loot1 - это удобная форма организации никла (см, далее), в принципе вместо

DO UNTIL МОЖНО бЫЛО бы ПРИМЕНИТЬ

и ио WHILE (см. ниже).

Вызов этой процедуры ножно вы полнить, например так: в

sum i, г. з,ч I или с другин количесгном на ранет--! ров: -. I

sum 1,2.x. Y, 2. 1256 I!

ес ни ваш список параметров со 1 стоит из строковых переменных, то!! в строке 130 надо ;>ыло бы принс и нить READ a$, d иг RF;AD a. а как! быть, если список сметанный и г.ч> IJ держит и числа и строки? и ,т>м1| случае nt'pe.i RKAD н,*до проверить. II очередной параметр с понояыо ITEM. () и использовать либо тот вари ант. либо другой, в строкОВЫХ не-реншшнх при этом можно избежать использования кавычек, ос:ли вм(;

сто BEAD ИСПОЛЬЗОВаТЬ READ LINE

(см. далее».

Обратите внимание HJ то, что список фактических параметров, передавемых через DATA и READ, исключает возможность им быть локальными, если вы специально это не зададите.

Рекурсия.

Процедура, которая вызывает саму себя, называется рекурсивной. Есть анекдот, что в одном из компьютерных словарей-справочников написали:

РЕКУРСИЯ СМ. РЕКУРСИЯ.

Рекурсия часто помогает очень элегантно избегать сложных программистских проблем, с другой стороны, это не самая быстро ра ботающая структура, к тону же возможны большие расходы памяти, ведь при каждом вызове процедурой самой себя Формируется новый временный список локальных перемен ных и для них выделяется память.

кроне того, несмотря на внешнюю простоту такой структуры, ее тюа-теяьвый раэбор может вызвать у программиста легкое головокружение, в качестве примера мы приведем процедуру, которая рисует бриллиант, а затеи еще один такой же. но поменьше. Внутри первого и еще один внутри второго и так далее, на минимальный размер наложено ограничение, иначе процесс йог бы продолжаться бесконечно, юо DEF FROC diamond я. У. size, diff

DEFAULT diff^15

PLOT X. Y. -3lZ6

DRAW -3]ze.size

DRAW size,size

DRAW size, -size

DRAW -size.-size 110 IK Size >* THEN

diamond n, v+size,slze-diff diamond x,v-slze. size-diff diamond x~size.y, size-diff diamond x>size,y,size-diff

130 END PROC

Вызвать ЭТУ процедуру ножно, например, так:

diamond 1гв,ее.ад

ОШИБКИ

Если вы попробуете вызвать процедуру, которую забыли задать. Вам дадут сообщение об ошибке W: "Hiasine DEF FROC". Если же Вы забудете закрыть процедуру с ло-мошью EKD PBOC - сообщение X: -но END PROC". программа будет стараться во вреия работы "перепрыгнуть" через блок, в котором задается процедура и не сиожет этого сделать.

Если при вызове задано больше фактических параметров, чем их есть в наличии в описании процедуры. - сообщение " Pararoet er error." Если тип формальных параметров не совпадает с типом реально установленных фактических параметров - "Nonsense in BASIC".

оператор END PROC может генерировать сообщение "Variable not found", если оказывается, что переменная. которую процедура должна передать в качестве выходного параметра, не существует.

ГЛАВА 4. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

Кроме заложенной в БЕТА ЕЕИСИ-Ке з. о концепции использования процедур, о чем мы только что написали, введены еще дополнительно некоторые средства, обеспечивающие возможность структурного программирования :

Операторы DO, LOOP. EXIT IF. WHILE и UNTIL обеспечивают те же дополнительные возможности организации ШПСЛ08. ЧТО И REPEAT И WHILE в более ранних версиях и еще больше повышают гибкость в программировании.

В структуре операторов IF. . . THEH можно использовать ELSE.

Оператор он позволяет выбрать необходимый номер строки, это как бы упрошенная форма операторов

CASE ИЛИ SWITCH,

команда LIST FORMAT может вам обеспечить вывод текста программы в структурированной Форме ("лесенкой") так. как это ПРИНЯТО в языках, поддерживаюимг структурное программирование (ПАСКАЛЬ-си и ДР.).

ГЛАВА 5. ОБРАБОТКА ДАННЫХ

В данной версии языка введены дополнительно следующие операторы, обеспечивающие обработку данных разного типа:

JOIH. COPY. DELETE, SORT - ДЛЯ

работы с массивами и со строкОВЫМИ переменными.

IHA RRAY И ISSTR1NG - Поисковые ФУНКЦИИ.

ФУНКЦИИ LEHGTH, CHAS», H1ЖПЕК.

операторы: EDIT <переменная>. SAVE DATA, USIBG.

ФУНКЦИИ: EOF. SHIFT», USIHG*.

JOIH и СОРУ позволят Вам перемешать или копировать массив данных или его часть в ДРУГОЙ массив. Теперь вы можете во время работы изменять размер массива, оператор DELETE позволяет удалить элемент массива, а SORT - выполнить его сортировку по алфавиту или по числу. эти же команды могут быть использованы и при работе со строковыми переменными.

функция IHARRAY выполняет просмотр массива и поиск в нем нужного вам элемента, то же самое делает IKSTRIHG. но для строковых -переменных. функция LENGTH выдает размер массива и его месторасположение, она может позволить Вам разделить массив на части и загрузить и обработать его по частям, если он слишком велик, чтобы поместиться в памяти компьютера в то время, как там при

сутствуб6т бета-бейсик 3. о. функции CHAR» и HUMBER Дают возкож-ность создавать "целые" массивы.

Теперь Вы можете редактировать [иэиеняты переменные в той же мере, как вы редактируете программные строки, все программные переменные можно отгружать на ленту единым блоком с помошью SAVE DATA. Форматирование данных

ножно выполнить с помошьк» USIHG или USING*. функция EOF (End Of

File - "конец Файла") служит для работы с никродраивом и может сигнализировать о том. что ввод данных из Файла завершен. Оператор SHIFT среди ПРОЧИХ дел выполняет и такую полезную операцию, как изменение регистра букв, КОТОРЫМИ записана строковая переменная.

ГЛАВА 6. ГРАФИКА

Используются следующие операторы и функции:

ALTER, CONTROL. CODES, CSJZE, DRAW TO. FILL, GET <область экрзна>. OVER a. PLOT. POKE, ROLL.

SCROLL. WINDOW. XOS/XRG/VOS/YFG, ФУНКЦИИ SIHE. COSE. FILLED.

HEHOEY*. 5CEЙ$,

Вот в ДВУК словах, для чего они предназначены і более подробно-мы рассмотрим каждый оператор и каждую функцию в бдижайша вьшге как!:

ALTER - позволяет гибко управлять оветошчи атрибутаки экрана.

DRAW TO - вычерчивание линий к заданной координат.?.

GET - сохраняет заданную область экрана в виде строковой переменной.

PLOT - восстанавливает на экране (в произвольной области) сохраненный с помошью GET *рзг мент.

C5IZE - с его помошь» вы можете увеличить или уменьшить размер фрагмента экрана, принятого с помошью GET перед тем, как восстанавливать его по PLOT.

POSE - допускает быстрые манидуляции с областями памяти..

FILL - заполняет область, экрана, наводящуюся внутри замкнутого контура, избранный цветом IHE или PAPER,

ROLL - переметение экрана или его части в заданной направлении.

SCROLL - то же самое, но с возвратом, когда например т<ч что ушло за левую Гранину экрана, начинает появляться справа.

SCRN» ~ распознает симво^ш графики пользователя.

WIHDOW - организация концепции "окон".

XOS. XRG. YOS, YRG - изменяет исходную координату экрана для графических ФУНКЦИИ и масштабный коэффициент по двум направлениям.

ГЛАВА 7. Сервисные и отладочные возножности

ALTER (.. ! - поиск и замена со тексту программы.

лито - автоматическая нумерация строк.

DEF KEY - этой конзняои Вы ножете задать до ЗБ операторов. функций или строковый сообщений на пользовательских клавишах.

DELETE - удаление блока програниных строк.

LIST/LLIST - распечатка текста программы или ее фрагмента в диапазоне от. . . до.

LIST/LUST DATA/VAL/VAL$ распечатка программных переменных.

LIST/LLIST DEF KEY - Распечатка определении, присвоенных пользователем назначенным клавишам.

L1ST/LLISR PROC - Распечатка текста процедуры.

LIST/LLIST KKF распечатка только тех строк, в которых есть оператор REF.

REF - поиск программных строк с этим оператором.

RENUM - перенумерация программного блока или кодирование.

HEMO эта функция возвращает доступный объем свободной нанята.

(Продолжение следует)

МИХАЙЛЕНКО В.С.

ЗАЩИТА ПРОГРАММ

Сегодня мы начинаем печатать объемный труд нашего читателя из г. Минска Михайленко В.С., посвященный вопросам защиты конпьютерных программ. Книга написана им специально для "ZX-РЕВЮ".

Мы очень рады, что лед тронулся и стали появляться крупные работы отечественных авторов, за те несколько лет. что мы работаем в области информационного обеспечения "Синклер"-совместимых машин, мы очень хорошо узвали, как много у нас талантливейших программистов, способных разработчиков. Но с другой стороны, мы хорошо знаем и то. как трудно их подвинуть на то, чтобы написать хорошую книгу, статью, цикл статей. "Сделаю все. что угодно, но писать книгу -разве что вод дулом пистолета". -вот типичный ответ нашего разработчика, значительно меньшая часть все-таки соглашается, причем охотно, но после первого но-рыва быстро охладевает и на этой всо кончается.

откуда же иы знаем о существовании этой армии талантов, если они ничего не пишут? - кожет спросить внимательный читатель, оказывается, все-таки пишут, они пишут интересные, еорошо проработанные, наполненные точный, конкретным содержанием критические письма на то, что читают в "ZX-РЕВЮ". кы благодарны км за это. но увы должны отметить, что собраться и написать пять-шесть страниц, блежушин идеями, нашему разработчику проше, чем собраться с духом к подготовить норкаяыш законченный материал, освещающий хотя бы одну идею, и который с интересом и вонаманиен будут читать миллионы начинающих поклонников "СПЕКТРУМа". а именно такой цифрой мы и оцениваем на сегодняшний день тех. кто уже включился в это всенародное движение.

Итак, мы предоставляем страницы В.С. Михайленко для освешения полезного, как нам кажется, вопроса защиты компьютерных программ. Ны полагаем, что кроме своей утилитарной дели то. о чем он пишет, имеет еще и большое методологическое значение, поскольку ьриоткрывает перед вами завесу над многими вопросами, связанными с системными особенностями "Спектрума".

Работа прошла техническое и литературное редактирование "ИНФОРКОМа" .

ЧАСТЬ 1

Система защиты компьютерных программ представляет собой исключение возможности просмотра текста программы и внесение в него изменения липом, ве уполномоченным на это автором программы.

Для компьютера "zx SPECTRUM" сушествует система мер по защите "бейсиковских" программ и программ в машинных кодах, этот цикл статей посвящается защите Бейсик-орогрзни.

Методов защити программ сушсствует очень много, но средн них можно выделить ряд направлении, по которым можно осуществлять подход к данной проблеме:

1. Исключить возможность остановки, (прерывания работы программы).
2. Сделать листинг программ нечитаемым.

ГЛАВА 1. Исключение возможности остановки программ.

1.1. Обшие рекомендации.

Самое первое, что необходимо сделать - это выполнить программ? автостартуюшей со строки с номером п. это выполняется при записи программы на ленту командой:

SAVE "ИМЯ" ЫКЕ П

Теперь после загрузки программа сана будет стартовать с указанной строки. Тем не менее, она может быть остановлена командой BREAK или по сообщению об ошибке.

Отключить клавишу BREAK можно. если в той строке, с которой UPO-исходит автостарт, поместить команду POKE 23513. PEEK гЗТЗО-5.

Кроне этого, возможно отключение клавиши BREAK за счет использования подпрограммы в машинных кодах, которая будет описана ниже.

Если же вы до каким-то причинам не желаете использовать подпрограммы в машинных кодах то необходимо помнить следуюшее.

Для хорошо отлаженной нрогрэикы остановку по ошибке можно исключить, надо знать, что чаше всего при взломе программы сознательно вносят ошибку во время исполнения оператора INPUT, например, если программа просит от пользователя ввести какое-либо число, он сознательно вводит букву. Программа останавливается с сообшеннем VARIABLE HOT FOUKD (переменной нет), чтобы избежать этого рекомендуется не использовать оператор INPUT, а организовывать опрос клавиатуры на основе функции IKKEY.

Если же Вам необходимо использование оператора INPUT (например при вводе многозначных чисел), то с целью защиты от прерывания можно сделать, чтобы программа либо "зависала$ (или саносбрасывалась ври остановке), либо вместо со-Обшения об ошибке осуществляла переход па заранее заданную строку.

Самосброс программы дает нс-пол ь зевание нижеприведенных команд. так. еслк поместить их в строке автостарта, то при попытке сделать BREAK или появлении сообщения об ошибке программа будет сбрасываться:

LET ERROE-256*FFEK J3614+

PEEK гзби;

POKE EKSOR, O: POKE ERROR +1,0

для обеспечения зависания программы используется размещение в стартовой строке команды: РОГЕ ?3659,O

Однако, пользователе компьютера 2K SPECTRUM наверняка заинтересует напомненный им оше раз Факт: ни в одной фирменной программе (имеется в виду рабочая программа, а не эагру::ч-нх) не происходит сбрасывания или зависания при нажатии на клавишу BREAK. Если после длительной загрузки программа "зависнет или сбросится, то это будет иметь недружественный эффект, гораздо эффективнее в этом плане использование специальных программ о машинных кодах, которые вместо выдачи сообщения об ошибке осуществляют переход на заранее заданный шаг программы.

Эти программы и методы их использования описаны в следующей статье.

1.2 Подпрограмма обработки сообщений с кодами D,H,L.

Нажатие клавиши BREAK во время исполнения программы (или же ей аналогичное STOP IK INPUT> в обы чнон рожиме может приводить к одному из следующих сообщений:

D: BREAK - COHT REPEATS

Клавиша BREAK нажата во время действия периферийной операции. Действие CONTINUE после этого оператора обычное, т. е. то, что указано в операторе.

н: STOP IK INPUT

некоторые введенные данные начинаются с оператора STOP, или была нажата INPUT L1HE.

действие CONTINUE обычное.

L: BREAK INTO PROGRAM (BREAK во время исполнения программы)

Нажата клавиша BREAK; это было обнаружено между исполнением двух операторов, строка и номер оператора в строке указывают на оператор, выполненный перед нажа тиеи BREAK, но CONTINUE переходит к следующему оператору.

Чтобы во время нажатия клавиши BREAK остановки не произошло, мо жет быть использована, например

подпрограмма в машинных кодах OH BREAK GO TO (подпрограмма Нбб из инструментального пакета SUFERCODE 5. 5).

Если Вы запустите эту подпрограмму со строки автостарта, то типы остановок В.н.L будут игнорироваться, а программа будет переходить к заранее заданному ванн номеру строки (Первоначально задана строка 9455).

Для тех. кто не располагает пакетом SUPERCODE 3.5. мы приводим дисассемблер

данной процедуры.

```
 ORG &0899
 60399 CD7C00            CALL 0124
 60902 3Б                DEC SP
 60903 3B                DEC SP
 60904 E!                POP HL
 60905 010FD0            LD BC, 15
 60908 09                ADD HL, BC
 60909 EB                EX DE, HL
 60910 2A3D5C            LD HL, (23613)
 60913 73                LD <HL).E
 60914 23                INC HL
 60915 72                LD (HL),D
 60916 C9                RET
 60917 76                HALT
 60916 CD8E02            CALL 654
 60921 7B                LD A.E
 60922 FEFF        CP 255
 60924 20FS        JR HZ.60918
 60926 3A3A5C            LD A, (23610)
 60929 FE0C CD 12
 60931 геол        JR z,еоэчз
 60933 FEI0        CP 16
 60935 2805        JR Z.60913
 60937 FE14        CP 20
 60939 2802        JR Z.60913
 60941 1819        JR 60968
 60943 3C                1HC A
 60941 32Й15C            LD (23681).A
 60947 FD3600FF          LD (IY*0),2bb
 60951 2M725             LD HL, 9495:
 загрузка номера строки пере хода в Бейсике.
 60954 22Ч25C            LD (236181,HL:
 60957 210000            LD HL,0
 60960 22445C            LD (23620), HL
 60963 3P                DEC SP
 60964 3B                DEC SP
 60965 C37D1B JP 7037 60968 C30313 JP 4867
```

наилучшим вариантом было бы. если бы Вам удалось набрать эту процедуру в Ассемблере и, откомпилировав ее, ПОЛУЧИТЬ соответствующую программу в кодак, для тех, кто не имеет такой вознож ности. ПРИВОЖУ программу на Бейсике, которая введет и эап/стит данную процедуру.

```
5 CLEAR 24999
10 FOR 1^25000 TO 25071
20 READ A: POKE I. A
30 NEXT I
40 DATA
205. 124.0,59,59.225, 1. 15. 0.9. 235.42,61,92, 115.35, 114.201, 116, J05. 142, 2, 123.
      254, 255. 32. 24в. 5в, 58.92
50 DATA
254. 12. 40. 10. 254. 16, 40, 6, 254, 20, 40, 2. 24, 25. 60, 50. 129, 92. 253. 54. 0,
      255. 33. 23, 37, 34. 66, 92. 33. 0, 0. 34. 68. 92. 59. 59. 195, 125. 27. 195, 3.
      19 ___"_________________
```

Данная Бейсик-программа позволяет вводить эту водцрогранму в любое место оперативной памяти. Теоретически возможны 3 варианта:

1) Ваша Бейсик-программа сана формирует эту процедуру и запускает ее, т. е. вышеприведенная Бейсик программа является началом Вайей исходной программы, которую необходимо защитить.

2) вы формируете в удобном месте эту процедуру в кодак, записываете ее из кассету в виде отдельного файла машинные кодов и потом, с помощью программы-загрузчика загружаете эти коды и запускаете.

3) вы совмещаете программу на Бейсике и эту программу в кодах с использованием

оператора REM методом который будет описан в одной из следующих глав и затем сразу после загрузки, непосредственно из Бейсика запускаете ЭТУ процедуру.

третий метод является самым удобным. К недостаткам первого относится то, что пока Бейсик будет формировать подпрограмму в кодах (перебрасывать числовой массив), взломщик ухе сумеет остановить программу. Второй метод усложняет ззгрузку: в самом деле. Вам придется, делать программу загрузчика, загружать кодь., за пускать их и лишь потом, обезо пасив себя от взлома, загружать свою основную программу.

1. 3. Подпрограмма обработки сообщении об ошибке

, кроме 0:OK; fl:END OF FILE; 9:STOP STATEMENT.

Нногие программы имеют возмож ность ввода ошибки, т. е. останавливаются из-за неправильной своей работы. В эток случае на экране дисплея индицируется сообщение об ошибке. Чтобы этого не происходило, а вместо выдачи сооыпений осуществлялся переход к заранее заданной строке Бейсик-программы, используется процедура: ON ERROR GO TO (процедура и 65 из программы SUPERCODE).

Ниже приведена Бейсик-программа, с помощью которой можно загрузить пропедуру :"ON ERROR GO то" в любое место оперативной памяти, в ноем примере загрузка осуществляется с адреса 25000.

```
10 CLEAR 24999
20 LET A-25000
30 FOR 1 = 0 TO 72
40 READ B
50 POKE a$I.B
60 NEXT I
70 REM ЗАПУСК ПРОГРАММЫ
80 RANDOMIZE USR 25000
90 DATA
205. 124.0.59,59.225. 1, 15.0,9 235. 42. 61. 92, 115. 35. 114, 201. 59, 59 205, 142. 2,
      123, 254. 255, 32, 248, 56. 56
100 DATA
92. 254. 255. 40, 33. 254, 1,40. 29. 254 8, 40. 25. 60, 50. 129,92.253,54,0 255, 33, 23,
      37. 34. 66. 92, 175, 50, 68
110 DATA |
92, 253, 203. 7, 254. 195. 125, 27, 51, Я
51, 195,3. 19 II
```

процедура "ON ERROR GO то" | имеет длину 73 байта и осляест | вляет переход к строке Бейсика! 9495 (допустимы изменения). В

Рассмотрим, для чего же уио-8 требляется эта зрограммя. В раз S деле 1. 1. было сказано, что одним* из методов взлома является ввод! несуществующей переменной внестоИ числа в операторе INPUT, На УРО-? вне Бейсика рекомендовалось ис-1 пользовать ОПРОС клавиатуры с по I мошьго INKEY*. Данная процедур^! самостоятельно анализирует подобного рода ошибки и осуществляет переход к строке 9495, . , При атом по адресу 23681 можно узнать код ошибки.

Точно такой же переход эта программа будет осуществлять, ее ли будет произведена попытка деления на о. оператор PRINT попытается распечатать символ с поно шью управляющего кода AT за пределами экрана; встретится RETUKH без GO SUB или NEXT без FOR и UP.

Наилучшим способом использования данной процедуры является со внешение ее с Бейсик-программой с использованием оператора REM методом, который будет ниже.

Чтобы подготовиться к его выполнению. Вам необходимо сформировать эту программу в произвольной области ОЗУ, используя Бейсик- программу данного раздела и записать ее на магнитофон в аиде отдельного файла CODE.

1.4. Метод защиты, основанный на совмещении Бейсика и программы в машинных кодах.

Многие пользователи компьютера ZX SPECTRUM бывают удивлены, когда после загрузки всего лишь одного Бейсик-файла программы на экране возникают эффекты,

которые на Бейсике создать заведомо не -возможно, такое же чувство возникает и тогда, когда после загрузки первого Бейсик файла дальнейшая загрузка ведется необычным способом, несмотря на то, что загружался именно Бейсик-файл [свидетельством этог'о является то, что Вы подали команду LOAD "").

Все дело в том. что в данном случае использован прием, позволяющий совместить в первом загружающемся файле как Бейсик-команды, так и машинные коды.

Рассмотрим, как это осуществить на практике, предположим, у нас имеется отлаженная программа в машинных кодах, имеющая длину п байтов, и мы желаем совместить ее с Бейсик-программой- (Данное совмещение накладывает ограничение на программу в кодах: она должна использовать индексную адресацию, переходы могут быть только относительными, т. е. необходимо, что бы эта программа работала, будучи загруженной в любое место оперативной памяти).

один из приемов состоит в тон. чтобы разместить программу в кодах после оператора REM. т. к. текст, следующий после этого оператора, не обрабатывается компьютером, (подобное совмеввние также накладывает ограничение за ис-пользование кодов в программе. ни один код этой программы не должен быть равен 13(DEC). OD(HEX). ПОСКОЛЬКУ в Бейсике этот код свидетельствует об окончании строки и символы, следующие за чин. обрабатываются как новая строка Бейсика!.

для того, чтобы осуществить совмещение, нан необходимо сформировать строку с оператором КЕН так, чтобы количество символов. следующих после REM. было равно ЧИСЛУ байтов программы в кодах, т. е п.

Для этого наберем с клавиатуры:

! ЯЕН LLLL. . . LLLL

Иной был использован символ L но в принципе можно использовать и любой ДРУГОЙ.

Как Вы ухе обратили вниманий, строка с оператором ЙЕН должна идти первой в программе, это не-обкояимо для того, чтобы легко можно было определить адрес, tio которому нужно грузить машинные коды и из которого вызывать их на исполнение.

Если Вы не изменяли системную переменную PROG, то она указывает на начало Бейсик программы по адресу гз?55.

следовательно, загрузку кодов нак необходимо начинать с адреса ?3760 (Это объясняется тем. что первые два кода Бейсик-строки характеризуют ее номер, следующие два - ее ДЛИНУ и пятый символ -это код оператора, в данном случае 8ЕН).

теперь заменим набранные нами символы в количестве п байтов на программу в машинных кодак аналогичной длины. Пододим команду

LOAD " CODE 23760. H

(В случае, если системная переменная PROG у вас смешена, тс Вам необходимо узнать адрес начала загрузки кодов, подав следуюшую команду с клавиатуры:

PRI8T (PEEK 23635 * 256 * PEEK 23636 +5)

теперь. при загрузке кодов, вместо адреса 23760 укажите адрес, который Вы получили, загрузив коды, мы добились поставленной пели. Теперь в Бейсике находится программа в кодах. которую при желании можно запустить, подав в одной из нижеследующих строк Бейсик-программы, команду RANDOMIZE USR 23760

В случае смешенной системной переменной PROG, вам будет необходимо видоизменить команду: RAJIDOHIZE USR (PEEK гзб35+25б«PEEK гзбзб + st Нетод совмещения чаше всего используется, когда вы хотите защитить наиболее уязвимые места

программы, которые вам наиболее дороги и вам бы не хотелось, чтобы они были кем-либо изменены. громе этого, такой подход позволяет загружать малинные коды без "заголовка" методом. который будет описан далее, что затрудняет вскрытие данного нашиннокодового блока. поскольку не известен его адрес.

Нак вы уже обратили внимание. обычно загружающийся программный блок состоит из двух частей, первая часть является тем заголовком. который содержит название программы, ее длину и адрес, начиная с которого она располагается в памяти компьютера, а вторая часть - это уже непосредственно коды, следует отметить, что аналогичную структуру инэет и Бейсик программа и массивы, но "заголовки" этих блоков имеют

принципиальные различия,

Среди наиболее любопытных особенностей метода скрывания машинного кода в строке KEH следует отметить тот факт, что после подачи команды LIST распечатка может быть остановлена с сообщением о неправильном цвете (IHVALIB COLOR), это объясняется тем, что в машинных кодак могут встретиться команды, код которых не может быть опознан компьютером в строке Бейсик-программы, и этот любопытный факт, совместно с использованием метода зануления. который будет описан в следующей главе, дает великолепную защиту от листинга. Необходимо, однако, предупредить читателя, что создавая программы в кодах, для их последующего размещения в Бейсике, необходимо сразу определиться в , адресах, т. е. точно установить адреса, где данная программа будет находиться и ра.^отать. чтобы не было сбоев из-за неправильной адресации.

Если Вы все же хотите использовать программу в кодах, которая была рассчитана на определенное место в памяти, то чтобы ничего не переделывать, вам просто необходимо дополнить ее размешенным в начале блоком переноса с использованием команды LDIR (более подробное описание этой команды дано в соответствующих методических разработках "Инфоркома").

1.5. Оригинальный метод защиты, использующийся при загрузке машинных кодов.

самые первые программы для ZX SPECTRUM отличались примитивноетью в плане защиты (да и не только защиты). Обычно они состояли из Бейсик-программы, управлявшей загрузкой, которая загружала коды и запускала их с определенного адреса. Взломщик ног беспрепятственно смотреть как Бейсик программу, так и программу в кодах.

Более поздние авторские разработки были гораздо лучве продуманы в отношении защиты.

Как известно. загружающаяся программа состоит из заголовка и самой программы. В заголовке находится описание типа программы: БеЙСИК. CODE. SCBEEH* и т. д. Кроме этого, таи находится название программы и адрес ь памяти, с которого начнется загрузка, а также длина программы (для программ в машинных кодах).

Чтобы скрыть эти данные, т. е. длину и адрес начала, разработчики создали систему загрузки. позволяющую загружать машинные коды без заголовка.

Это делается с использованием специальной процедуры в кодак, которая в некоторых случаях совмещается с программой на Бейсике с использованием оператора KEM.

К программам. использующим данный метод загрузки, относятся HAT П. KAKBO III И ДР.

Рассмотрим, как этот приен реализован программе HAT II. Ниже приведен листинг загрузчика.

```
J REM GKACKED BY HIHALLEMKO VS 1991
10 FOR ti-0 TO 12: READ a:
POKE 64000+n,a 20 NEXT n 30 DATA 55. Ь2,2Ь5. 221. 33,0.64.
!7.254.27. 195 86,5 40 RANDOHIZE USR 64000
```

Фактически Бейсик-программа служит лишь для того, чтобы сформировать и запустить процедуру в машинных кодах. Запуск осушест нляется в строке to, а строки ю...зо ответственны за форннрование этой Ш'оиедуры, начиная с адреса &4COO ниже приведен дне ассемблированный текст данной процедуры.

```
64000 SCF
64001           LD A. 255
64003           LD IX, 16384
64007           LD DE, 7166
64010 JP 1366
```

действует эта программа следу юшин образом, в регистр IX загружается адрес начала загрузки ко дов, в регистр DE - обшая длина программы в кодах. остальная система команд необходима для правильной работы встроенной в ПЗУ программы загрузчика с ленты. Перевод на ЭТУ программу осуществляется в строке &4010. (Все ис пользуемые величины -

десятирич ные).

этот тип загрузки используется во многих программах. В част ности. программа GREEN BE RET ис пользует встроенную процедуру за ГРУЗКИ как nonпрограмму, вызывая ее по CALL 1366 для нагрузки не скольких блоков кодов.

Следует добавить, что аналогичным образом действует загрузка в программе RArfBO m. отличи тельной ее особенностью является загрузка кодов, занимающих всю оперативную нанять. Никаким ДРУгни способен здесь загруэху осуществить GU не удалось.

1,6. Метод защиты, используемый в программе FIST III.

Защита, используемая з программа FIST III. достаточно проста и не является идеально надежной, но. в то же время, об этой системе команд надо кто знает и поэтому в некоторых случаях она ножет оказаться достаточно эффективной.

по налей классификации этот тип защиты относится к разряду тех. которое делает листинг программы не читаете* для этих целей бордюр скрашивается в черный цвет и подается система команд:

BCRDEK O: POKE 23624,0:

POKE 23570,16

Этот метод защиты легко можно разблокировать, подав команды:

BORDER 7:POKE 23570,6

кроме защиты, эта программа имеет еще одну особенность, на которую могли не обратить внимание пользователя: заставка в середине экрана формируется на Бейсике, создана она довольно оригинальным способом, ЕО самым любопытным является то, что системная переменная, используемая в этоя программе, в фирменном описании компьютера (автор - вик-керс) охарактеризована, как неис пользуюшаяся,

ниже приведем текст программы, после выполнения которой в середине экрана очень крупным шрифтом появляется надпись: G00D LUCK то YOU

10 ЙЕН CRACKED B? HIHAILEBKO

VADIM HIHSK 1991 2O FOR 1-7E TO 79 25 POKE 23681,1 30 LP8INT "G00D LUCK TO YOU" 40 NEXT 1

В данной случае автор» используют команду вывода на печатавшее устройство LFRINT для поэтапного высвечивания на экране вышеуказанной нашШСИ.

Если вы хотите получить подобное изображение не в середине, а в верхней части экрана, то Вам необходимо подобрать соответству-каган образом изменение нереиен-ной 1.

Лобоагтаал эффект получается вря исвользоваля яэмтмшя

го и» 1=тг то 78 м т. ж.

ГЛАВА 2. Методы защиты от листинга или как сделать текст программы нечитаемым.

Данная глава посвящается изучению методов зажиты от листинга, применяемых в большинстве фирменных программ, в частности в их загрузчиках, написанных на Бейсике, либо так или иначе использующих Бейсик. Кроме этого, рассмотрены некоторые методы, применяемые 'взломщиками* компьютерных программ, т. я. Езккерамн. в частности Билом Гилбертом (Bill Gilbert - если человэк с такнн именем действительно существует, а не нсаольэует йсзвдоннн): PRGAZ

SOFTWARE (Т. Н. КОРПЭРЭПИЯ ПО взлому, которая даже оставляет свой телефон); ROBV CRACKING SEKVISE, нсаользухжен з своей арсенале достаточно примитивные приемы в отличие от того же Билла пшберта и ПР.

3 связи с этим я обращаюсь ко всем читателям данного материала с просьбой не использовать описанные иной приемы взломщиков для установки в созданных не Ванн программах ваших надписей и т. п., поскольку это противоречит элементарным этическим норнам, не говоря уже о порядочности.

После длительной работы я могу сказать с уверенностью, что в компьютере zx SFECTRUH нельзя создать полностью защищенных программ. Это налагает большую

ответственность на человека, они -снваювшго приемы взломщиков, я ничего не скрывал в изложении данного материала, полагаясь на порядочность и внутреннюю культуру читателей, призываю Вас использовать данные приемы исключительно для защиты своих программ, а также для совершенствование своей текинки программирования, вы можете использовать описанные ниже рекомендации для входа в программу с пель» изучения новых приемов программирования, но к тем, кто ис-используэует, их для того, чтобы калечить чтжяе программы, не умея написать свои, надо относиться также, как мы отнеслись бы к человеку. который на нашив глазах лазит по чужим карананам-

2. 1. Общие рекомендации.

Простейжяй прием сокрытия листинга состоит в том, чтобы сделать одинаковыми цвета символов IIMK) И ФОНа (PAPER).

в тех местах, где программа должна сделать вывод на экран в операторе PRI1T вставляет в качестве юемеша нравкдьзЕне цвета.

Например, зажав ввачаде ПРОПАШИ) костят:

ю ID т: ?лпк т

м нем ВМАХШЯШКОСТШ асаодь-зуем нотиьвый цвет:

20 PBIIT IXK о; -zx SPKcrauii-

ДРУГОЙ приек состоит в искажении набора символов путем задания -Фальшивого-значения системной переменной CHARS, для примера наберем с клавиатуры. POKE 2360Б.8:РЯ1НТ "ZX SPECTRUM": POKE 23606.0.

Небольшое отступление: Краткое описание системной переменной CHASS.

Как известно, системная переменная CHAES ответственна ЗА место раеполоаение шрифта в инк "SPSCTKUH". в своей обычном содержании она указывает на адрес шрифта, защитого в ПЗУ, а точнее

- на тот адрес. КОТОРЫЙ находится на гэб байтов ниже, это легко проверить:

PKINT PEEK 236О& PRINT PEEK Й3607

Точный адрес можно узнать, подав с клавиатуры клмандг:

PRIKT PEEK гЗБОб * 2Г)5*PEEK 23607 +856

Вы получите 15616. ОРИ этом по адресу 23606 находится О, а по адресу 236С7 - число 60.

среди практических использования этой системной переменной следует отметить возможность создания своих шрифтов в т. ч. русского и других национальный, а также переключение шрифтов с одного на другой, подробно эта технология разобрала в разработке

•Большие возможности вашего -спкктруна", выпушенной ИНФОРКОМОМ В 1989 Г.

в окончании комментария привожу любопытный пример использования иною этой системной переменной, нногие из Вас, вероятно, видели, что при загрузке некоторых Фирменных программ используются счетчики, они могут бкть разяич-ны. Либо это постоянно укорачивавшаяся линия [уменьшение длин» свидетельствует о процессе за-ГРУЗКИ, а сана длина показывает непосредственно сколько загрузилось и сколько осталось), использованная, в частности в программе SPLITTING, а также цифровые счетчики, имитирующие механические, подобные тем, что показывают пробег в автомобиле.

Кроме того, такая надпис ь. инитирухшая механический счетчик, использована во время демонстрации в программе FIST ill (режим DEHOHSTRATIQS, который появляется после загрузки, если не нажинать никакие клавши).

Подобный имитатор механического счетчика ножяо создать и вам. гели использовать возможности си-стенной переменной CHARS.

наберите предложенную ниже Барограмму и запустите ее:

```
10 POK 1=60 TO 0 STEP -1
20 POKE 23606.i:PAUSE 10
30 PRIIT AT И.1Ы-0-
40 Ю5XT 1
```

Подобный очень любопытный эффект наблюдается потому, что за каждый никл операторов FOK-XEXT процедурой, ответственной за ое-чать. распечатываются 8 байтов

предполагаемого знака о (в данном случае). но с каждым проходок системная переменная все ближе подходит к своену истинному значению, а за счет того, что шаблоны ОЯФР в ПЗУ расположены рядом, создается впечатление, что цифры следуют одна за другой по возрастанию или убыванию.

схематически этот процесс нож но представить так:

Рис. 1

Рис. 2

і этой с :еме горизонтально расположены байты шаблонов символов, хранящиеся в ПЗУ. и, в зависимости от того, на какой адрес указывает системная CHARS, в области, появлявшейся на экра не. (на экране появляется одно знакоместо 8X8). мы видим ОПУ скажшиеся вниз цифры, имитирующие движение механического счетчика, до тек ПОР. пока они не достигнут необходимой нам вели чюш - в моем примере о.

Итак. системная переменная СНЛК5 может быть использована для защиты от листинга в Бейсик программах. Если адресовать ее и те области памяти, где вообше ничего нет. например. POKE Н3б07,гоо. то все символы будут выглядеть, как пробелы и на экране вообше ничего не будет, вам же необходимо не ред всяким оператором PRINT включать нормальный режим и выключать его после PRINT.

?., Е. Универсальная система защита - метод нулевых строк. почти во всех программах к zx SPECT8UH присутствует нулевая строка. В большинстве программ она выполняет следующие функции:

1) Залита Йашинных кодов, раз неиеннык после -оператора REM от редактиро вания.

2) Зажита от листинга той час ти Бейсика, который размевен в нулевых строках.

3) Залита от редактирования надписей, оставляемых фирмами и взломщиками.

для того, чтобы сделать первую строку программы нулевой, необхо дико подать команды:

POKE гз755.о

POKE 33756.О

Образующаяся строка имеет номер о. она реально обрабатывается программой но не поддается редактированию, т. е. ее невозможно вы-звать командой EDIT.

Предлагаемая вашему вниманию универсальная система запиты включает в себя "зануление" всея строк программы (разумеется, этот метод действенен только для программ, где отсутствуют команды условных и безусловных переходов - GOTO, GO SUB и наибольшую эффективность приобретает для веболь ших программ, управляющих загрузкой) . Это становится возможным, т.к. программа выполняет все ко манды последовательно, а номера строк использует только для команд-переходов, преимуществ же у этого метода довольно много: при удачном сочетании его с методом защиты от листинга, использующим управляющие коды (подробно описан в следующей статье), это позволяет сделать текст вашей программы полностью нечитаемым.

Действительно, абсолютное "за-нуление" не дает возможности гка-эать оператором LIST на какой-либо конкретный шаг в программе. при подаче же команды LIST о. первая же строка программы с до-мошью использованных там управляющих кодов сделает одинаковым цвет INK и PAPER, либо закроет текст программы от чтения другим способом.

Иною разработано два метода "зануления". первый выполняется вручную, с непосредственным Вашим участием. Второй - делает все аи тематически, используя специально созданную для этих целей прогряиму. Предпочтительно, на ной взгляд, использовать первый нетод. Поскольку это позволяет Вам лучше узнать структуру Ьгйсикл и

быстрее освоить принципы работы компьютера. Но. тем не менее, в этой статье мною будут рассмотри мы оба подкопа.

для того, чтобы правильно ПРО-

обкодимо четко представлять себе структуру Бейсик строки, которую условно можно представить в виде:

НН NH <текст строки> ENTER, где:

МН - номер строки, описан

ДВУМЯ числами: NN - длина строки, описана

тоже ДВУМЯ числами. В интерпретаторе Бейсика под номер строки отводятся два знака, причем первым идет старший байт номера, а вторым младший. (для тек, кто интересуется. почему соблюдается именно такая последовательность, рекомендую прочитать подробную информацию об этом в трехтомнике "ИНФОРКОМа"первые шаги в машинных кодах).

Текст строки совпадает с Вашим исходным текстом, за исключением того, что числа представлены в несколько ином виде. [Более подробно об этой и иных системах представления чисел читайте в том

же трехтомнике в т. 1 на с. 56>. Заверилот строку условный кол оператора ENTER - I3(тринадцать».

Для того, чтобы сделать номера строк Вашей программы нулевыми, необходимо просмотреть каждую ячейку памяти Бейсика и. зная СТРУКТУРУ Бейсик-строки, опреде лить, в каких адресах памяти на-холится код номера строки, запи сать номера этих ячеек, после че -го с клавиатуры заслать в эти ячейки о командой POKE,

для получения дампинга всех ячеек памяти Бейсика, необходимо использовать небольшую программу, которая размешается в самом конце Бейсика:

```
9997 FOR 1-Й37Э5 TO Б5000
9998 PRINT 1ITAB 7:PEEK 1; TAB 11; CHR$ (PEEK 1)
9999 NEXT 1
```

Теперь запустим эту программу командой GOTO 9997, после чего на экране будет печататься дампинг Вашей программы (предлагаемые мною строки должны быть набраны после того, как в памяти уже на холится Ваша программа) в следу юшен Формате:

- номер ячейки памяти:

- десятичное значение числа, содержащегося там; символ, соответствующий дан НОМУ числу.

В колонке символов, соответ ствуюших содержимому ячеек. Вы увидите приблизительный листинг своей программы. Вашей задачей является обнаружить окончание данной строки Бейсика. Сеиде тельствон этого служит код ENTER

!3. Причем. если десятичное значение 13 распечатается на эк ране, то символ ПО?» 1J является непечатным дли /х :>ркстки и кон пьютер ОСУШСМ.'ТНИТ не и? к од на сл<? дуюиою строю', обнаружим, т.жим образом, НОМОР ячейки, в которой содержится код 13. свидетсльстну юший об окончании строки Бейсика, Вы записываете номера следующих за ним ячеек памяти, чтобы потом заслать в них о командой COKE.

Следует отметить, что Вам не• обходимо строго следить за тен. чтобы не превратить и 0 номера строк, осуществляющих "дамлипг ячеек памяти". После того, как Вы завершите - зануление", Вам необ ходимо уничтожить строки 9997. . . 9999.

Необходимо подчеркнуть ОДНУ маленькую деталь, помогавшую об легчить работу г программой рас печатки содержимого ячеек. Если это программа но какой либо при чине остановилась с сообщением об ошиске типа INVALID COLOR; NUHBKR T00 BIG и т. д. . то вам достаточно набрать с клавиатуры NEXT I и "даниинг* продолжится.

Второй метод заключается в ис пользовании программы для "зану леиия". Тот процесс, который Вы делали вручную, т, е. анализ СТРО ки и -зануление" ее номера. ЭТА программа делает самостоятельно:

```
9990 КЕН (С) ПРОГРаМНИСТ
нихайленко Вадим
НИНСК, НРТИ. ГР 010Е07
Беларусь. 1991
```

```
9991 FOR 1=г375B TO 65000
9992 IF PEEK 1=13 THEH IF FEES. (1+1>=39 AH» РЕИ (1+2)=6 THEH STOP
9993 IF PEEK 1=13 THEM
FOEEtl+ll.O: POKE (1*2),0: LET 1=1+*
9994 NEXT 1
```

Эта программа также размг ает-ся в конле Бейсик-файла ^десь необходимо строгое соблкг,:ме номеров строк) и осуществляет перенумерацию всех стоящих вереД ней строк в о. После работа эту проррамну также необходимо уничтожить. Действует же она по следушену приношу:

- в шпсле идет анализ Валей исходной программы, причем при обнаружении конца строки (об этом с видетедьствует код символа ENTER - снк*(13)). номер следу-юней за вея строки зануляется.

Строка 999S осуществляет контроль таким образом, чтобы не допустить превращения в о (ноль) номеров строк зануляюшей программы. (они находятся по номеру 9990. поэтому наличиэ этой строки - обязательно), после завершения работа программа останавливается оператором STOP.

это программа, однако, не сможет обратить в ноль самую первую строку вашей исходной программы. для того, чтобы и ее обратить в ноль, ван придется подать команду с клавиатуры:

```
POKE гзтээ,о
POKE 23756,0
```

Теперь ваша программа защищена: ни одну из ее строк невозможно вызвать для редактирования.

2.3. Использование управляющих кодов.

Одной из малоизвестных и надо исследованных возможностей компьютера является использование в Zx SPECTRUM управляющих кодов, в этом разделе, включающей в себе несколько статей, я опишу применение их для защиты, для использования в Бейсик программах с целью экономии оперативной памяти. а также применение их для создания оригинального хэддера.

Но для начала я постараюсь ввести читателя в КУРС дела, более подробно о знакомить е по с накопленным опытом использования управляющих символов.

К Я И

ИНФОРКОМ рекомендует Вам также обратиться к разделу "Маленькие Хитрости" в "ZX-РЕВЮ-91", где на с. Иб и 140-142 достаточно сод-робяо разбирался вопрос исподьзо-ваншя управляющих кодов в операТОР* PRI8T.

* « i

2.3.1. Введение

Не полностью изучив возможности своих персональны» компьютеров, многие пользователи бн&аст шомшрооаан. когда остановив загрузку, чтобы посмотреть содержание фирменной программы, вместо первой строки Бейсика они видят сплошную строку (без номера) с какой-либо надписью, например:

CRACKED BY BILL GILBERT () ГОД

Здесь может присутствовать также название программы или надписи иного рода (BILL GILBERT взят для примера - разумеется, другие "хэккеры* тоже используют управляющие коды). Текст же самой программы увидеть не удается.

С помощью управляющих кодов можно сметать текст программы в любом направлении в строке, располагать его в любом знакоместе экрана, управлять цветов символов и «она. регулировать мерцание и создавать повышенную яркость, включать инверсный режим и даже накладывать один символ на другой.

Кроне этого, специально для печати сообщений существует код. позволяющий пропускать сразу 16 символов в строке, т. е. он один как бы заменяет 16 символов типа "пробел".

знание принципов работы управляющих кодов позволит вам создать строку, аналогичную описанной в начале, а также так устанавливать цвета, чтобы текста программы

не было видно, кроне этого, управляющие коды можно использовать в "хэдерах" - заголовках файлов -а это позволяет стереть слово FBOGRAH ОРИ загрузке, либо же расположить название программы в любой точке экрана, либо сделать и то и другое одновременно.

К таким приемам в работе с файлами прибегали многие взломщики фирменных программ (с н. EXOLOH, GAHEOVER И Т. Д. J.

2.3.2. Самый первый шаг

Рассмотрим формирование строки без видимого номера и каких-либо побочных бейсиковских надписей. Суть этого процесса заключается в тон. что с помощью управляющих символов мы смещаем желаемую надпись влево таким образом, чтобы она перекрывала как номер строки. так и бейсиковский оператор (обычно в подобных случаях используется оператор ЕЕШ.

Изучим конкретный пример. Ны хотим, чтобы после остановки программы и подачи команды LIST у нас сверху появлялась надпись (без номера, указывающего, что это строка Бейсик-программы и без каких-либо операторов):

```
GOOD LUCK TO TOO YOUNG CRACK.
```

что в переводе с английского означает: УДАЧИ ТЕБЕ 10НЫЙ ВЭЛМШЯК. для этого наберем Бейсик-строку:

```
1 KEH LLLLLLLLLGOOD LUCE TO YOU YOUBG CRACK
```

Нежду оператором REM и десятью символами L (символ L может быть заменен любым другим, принципиального значения это не имеет), а также между L и основной над-ансью Вы не должны делать пробелов. Сейчас мы имеем "полуфабрикат", который будем изменять в соответствии с ранее поставленной целью, для этого четко определим последовательность наших действий:

1) сначала сместим особым методом исходную надпись на шесть знакомест влево.

2) зададим цвет INK исходной надписи,

3) Зададим цвет PAPEE для исходной надписи.

Будем последовательно выполнят эти пункты:

1. Для того, чтобы сместить надпись на & знакомест влево. Вам необходимо заслать управляющие символы оператора "курсор влево" вместо б символов L, для этого наберем с клавиатуры

```
FOR 1=23760 TO 237Б5:
POKE 1.8:
NEXT 1
```

Теперь, если вы без ошибок набрали исходную строку с оператором REM и правильно дали команду с клавиатуры, то при подаче команды LIST вы должны увидеть на экране надпись, содержащую 4 символа L и основную надпись, но уже без номера строки и без оператора

REM.

Выполним второй пункт:

2. Возможно, что зашишая свою программу, вы пожелаете изменить цвета PAPER и INK. но нам необходимо, чтобы в любых условиях ваша надпись была видна.

для этих целей необходимо задать цвета непосредственно для надписи. Делается это с помощью пряных команд, засылающих вместо символов L коды управления цветом:

```
POKE 23766.16
POKE 23767. o
```

в ячейку 23766 мы засылаем код управления цветом тс. а в ячейку г3767 непосредственно значение пвета. В данном случае символы будут иметь черный цвет INK.

после этой операций на экране должны быть видны два символа L и следующая за ними основная надпись.

3. изменим цвет PAPER. это производится точно таким ж способом:

```
POKE гзтев. 17
POKE 23769,7
```

в данном случае в ячейку с адресом 237&6 мв занесли код управления цветом PAPER, а в ячейку 23769 непосредственно числовое значение пвета Фона, теперь надпись,

появившаяся на экране после подачи команды LIST, будет содержать только то, что мы желали показать.

А если мы теперь подадим прямые команды

IKE O: PAPER O: BORDER O: CLS то после подачи команды LIST получим картину, аналогичную той. которую можно наблюдать после остановки многих фирменных программ.

Теперь давайте проанализируем, что же У нас подучилось из бывшей изначально достаточно простоя Бейсик-строки.

г3iv5 ? о в этих ячейках
23756 ? 1 распод. нон. строки
23757 ? 44 в этих ячейках
23758 ? о распол, шиша строки
J3759 REM 234 ЗДБСЬ НЗХОД. КОД
оператора REM
23760 в здесь расположен УПР.
23761 в код символа КУРСОР
23762 8 ВЛЕВО
23763 8
23764 8
23765 8
23766 16 здесь находятся УПР.
23767 о коды управления светом INK.
23Т6Й 17 здесь расположены
23769 7 УПР. КОДЫ. ОТВвТ-
ственные за цвет
PAPER
23770 здесь расположен наш . . . исходный текст: G00D
23797 LOCK TO YOU YOUHG CRACK

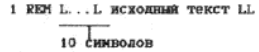
Следует предупредить читателя. что для того, чтобы выполнить все вышеизложенное необходимо, чтобы системная переменная PROG, расположенная по адресу 23635-г3б3б указывала начало Бейсик-строки с адреса 23755. это будет всегда, если вы не изменяли его (следует отметить, что это значение может изменяться само при подключении некоторой переферяи. в частности интерфейсов внешних устройств. например zx-INTBKFACE-1).

Рассмотрим в качестве следующего примера еще один, достаточно часто используемый хэкхерами прием, например, для того, чтобы сделать дальнейший текст программы (расположенный после строки, которую мы оформили в предыдушен примере > нечитаемым, необходимо сделать так. чтобы компьютер после подачи команды LIST сам останавливал распечатку в необходимом вам месте, это можно сделать несколькими способами, в качестве примера рассмотрим один из них.

после того, как компьютер распечатает строку с приветствием для вэдонликов. он должен получить команду (опять-таки с по-мошью управляющих символов) о тон. что задаваемый нами цвет -неправильный. Если такая информация будет получена, то распечатка остановится с выдачей со-обшения IHVALID COLOR (неправильный овет).

Для остановки распечатки в необходимом месте, аан понадобится зарезервировать еще 2 символа в том •полуфабрикате". который мы использовали в примере.

Следовательно, строка Бейснк-програкны в первоначальном варианте будет иметь вид:



для того, чтобы остановить распечатку, необходимо заменить ПЭРУ символов LL. расположенных после исходной надписи, на управляющие коды, подобрав их значения таким образом, чтобы они создавали заведомо несуществующую комбинацию цветов.

Чтобы узнать адрес ячейки, в которой расположен последний символ основной надписи, нам необходимо к последнему значению адреса перед этой надписью (в данном случае г576-9) прибавить количество символов, содержащихся в данной надписи, в надписи G00D LUCK TO YOU YOUHG CRACK соярежится 28 символов (мы должны считать не только

буквы, но и пробелы), следовательно, теперь мы можем установить адрес ячейки, в которой находите я код первой буквы L. первой из тех двух, которыми мы дополнили нашу надпись:

23769+гв=г3797

Адрес 23797 указывает на последний символ нашей надписи, следовательно, если вы не делали между основной надписью и парой символов LL пробел, ячейка, содержащая первое L должна иметь следующей номер:

гз797*1=г3798

проверим это. подав команду

PRIET CHR$(FEEK 23798).

На экране должен появитьс в символ L. а для того, чтобы убедиться что мы нашли адрес ячейки именно первого L. а не ВТОРОГО. подадим команду:

PRINT CHH*(PEEK 23799)

На экране должно тоже появиться L теперь нам нужно вместо первого символа L заслать код управления цветом INK:

POKE 23798.16

После этого компьютер, анализируя данные, поступающие за управляющий символон. теоретически должен будет выполнить следующую команду:

INK 76

(т.к. код символа L - 76), но это невозможно и компьютер остановит распечатку, выдав сообщение о неправильной паете:

IHVALID COLOR

таблица I

	0	1	г	3	4	5	б	7	в	9
0					TRUE VIDEO	IHVVID EO	PRINTз анятая	EDIT	КУРСО Р<--	ЖУР-СОР -->
10	ЮТ-СОРвн из	КУРСО Рвверх	DELETE	ЕЯ-TER	ЧИСЛО	режим GRAPH	УПР. IDE	УПР. PAPER	УПР. FLASH	УПР. BRIGHT
20	VHP.11 VEKSE	УПР. OVER	УПР. АТ	УПР TAB						

Описание управляющих кодов СПЕКТРУМА

Таблица 2

код наименование	Назначение	
о. 1 нет	используется после " "INVERSE-. -FLASH". -BRIGHT',	IK". "PAPER", "OVER". 'АТ- ИДИ "TAB".
г- 5 нет	используется толькоИДИ "TAB".	после ' INK". "PAPER". "AT"
6 COMMA CONTROL	выполняет табуляциюстроки.	в первую позвоню половины
7 EDIT	код. создаваемый ПРО клавиатуры при кажлч печатаемого символа используется как "ВЕ и терминалах.	граимой ввода с«И "CAPS SHIFT- И "1". не имеет, часто ЗХ"- код на принтерах

Бели Вы заблаговременно -за нулнли* строки своей основной программы а также - запулили" первую ее строку, т. е. ту. которая формирует данную надпись, подав команды

РОЖЕ 23756.О:POKE 23755,0

То теперь, подав команду LIST вы получите на экране сообщение, которое составляло

нашу исходную надпись, а продолжить распечатку никак не сможете т. е. невозможно указать на доследующие строки Зашей программы с помошыо оператора LIST - все они имеют нулевой номер.

Команда же LIST о снова распечатает строку с исходный сообяге -ниеи.

В примере 2 для того, чтобы скрыть дальнейший листинг. мы создали заведомо неправильный цвет, но можно поступить иначе, например, сделать после исходной надписи одинаковыми цвета 1НГ и PAPER. Это обеспечит не меньшую надежность в сокрытии дастинга. пододим команды:

```
POKE 23796.16
POKE г 37 99. 7
```

И. в тоже время, ваша защита судет иметь серьезный вид. если Вы вообше уберете с экрана какие-либо надписи, т. е. после подачи команды LIST экран будет продолжать оставаться чистый, это может быть достигнуто в тон случае, если мы сделаем одинаковыми цвета символов и фона еще до исходной надписи (строго говоря. необходимость надписи в этом случае отпадает].

Это делается после смешения строки на 6 пикселей влево (си. пример 1 ПУНКТ 1) подачей команд:

```
POKE 23765, 16
POKE 23767.7
POKE 23768. 17
POKE 23769. 7
```

На этой наш беглый обзор применения управляющих кодов можно закончить, во в последующих статьях материал будет рассмотрен более детально, вместе с теи. чтобы вы начинали не с нуля, рекомендуется проделать описанные эксперименты на компьютере, это будет способствовать лучшему пониманию материала.

2.3.3. Управляющие коды ZX SPECTRUM.

Управляющие коды в zx SPECTRUM используются так же. как и в других компьютерах для управления печатью на экране и на принтера, ниже приведены таблицы 1 и 2.

Внимательно изучив предлагаемый материал, вы сможете полноценно использовать управляющие коды в своих разработках.

(Продолжение следует)

Таблица 2 (продолжение)

код	наименование	назначение
в	BACK SPACE	код. создаваемый программой ввода с клавиатуры при нажатии "CAPS SHIFT" и "5". передвигает курсор влево. Воспринимается большинством принтеров, т.к. это ASCII код перехода назад.
9	RIGHTSPACE	Код, генерируемый программой ввода с клавиатуры при нажатии "CAPS SHIFT" и "B", код пере метения курсора вправо, но при его использовании позиция печати не меняется. ASCII код табулированной печати строки.
10	DAWN SPACE	как и вышеприведенное, но для "CAPS SHIFT" и "6". это ASCII код заполнения строки.
П	UPSPACE	как и вышеприведенное, но для "CAPS SHIFT" и "7*. это ASCII код для смешения печатаемой позипии вверх.
12	DELETE	как и выше приведенное, но для "CAPS SHIFT" и "0". Это ASCII код заполнения формата.
13	ENTER	Генерируется при нажатии клавиши "ENTER", выполняет возврат каретки и заканчивает печать строки при выводе на принтер, это ASCI I код перевода каретки. предшествует номеру строки в программе.
14		Используется для чисел в 5-ти байтной«ОРМе. ЭТО ASCII КОД 'SHIFT OUT'.

15		в стандартном БЕНСИКе не используется, это ASCII код "SHIFT IH". ножет использоваться в расширениях БЕИСика. например в БЕТА - БЕЙСИК е 3. 0.
16	IHE COHTFOL	Управляющий код иск. используется перед кодом, содержащий численное значение цвета "IBS". следует отметить что в качестве кода численного значения цвета используется не ASCII код 2. а значение 2. Например, для изменения цвета печати на экране всех символов печати аа красный, вам нужно использовать процедуру RST 16 с 16 в регистреА И ДВОЙКОЙ.
17	PAPER CONTROL	Управляющий код PAPER, используется аналогично коду INK. значения, следуете за ним, соответствуют стандартным значениям цветов"СПЕКТРУИА".
10	FLASH CONTROL	Как и вышеизложенное, но для FbASH. Код ножет быть соответственно только 0 или 1.
19	BRIGHT CONTROL	Аналогично вышеприведенному для "FLASH" только в данном случае используется для"SIGHT".
20	INVERSE CONTROL	как и вышеприведенное, но для "INVERSE".
21	OVER CONTROL	как и вышеприведенное, но для "OVER-.
22	AT CONTROL	код контроля -AT". который должен сопровождаться номером строки и столбца соответственно.
23	TAB CONTROL	как и вышеприведенное, но для "TAB", в данном случае код должен сопровождаться только значением столбца.

Дж. Хардман, Э.Хьюзон Перевод Тихоновой Н. А.

40 ЛУЧШИХ ПРОЦЕДУР

От переводчика

Данная книга является сокращеннын переводом книги "40 Best Hachlne Code Routines For The ZX Spectrun vith expianatory Test', j.Hardman * A. Hew son. со-держакей в себе набор программ в машиннин кодах с веська подробными разъяснениями принципов их работы.

Значительным сокращениям подверглись те раздали оригинала, в которых рассматриваются вопросы компьютерной терминологии и система команд zeo - советуем обратиться к трехтомнику по программированию в машинных кодах, вы-лэтвеиному "ИНФОРКОМОМ".

При подготовке данного пособия были также исключены описания программ, которые не представляют, на наш взгляд, особого ияте гее а.

РАЗДЕЛ А

1. ВВЕДЕНИЕ

Цель этой книги - обеспечить как начинавшего, так и опытного пользователя компьютера ZX SFECTRUH полезными. интересными и развлекательными программами в машинных кодах.

Книга имеет 2 раздела.

Раздел А описывает те особенности SPECTRUMa. которые важны при программировании в машинных кодах, некоторые процедуры системного ПЗУ. а также СТРУКТУРУ машинного языка.

Раздел в представляет сами программы. Они представлены в стандартном формате. который детально описан в начале раздела, программы полностью закончены, т. е. они могут быть загружены и использованы индивидуально, без обращении к другим программам.

Предлагаемые программы могут быть загружены с помощыо простого загрузчика машинных кодов (ПС LOADER - наш. ЗАГРУЗЧИК), описанного в начале раздела в.

Общие сведения о БЕЙСИКЕ и машинных кодах

Микропроцессор Z80A. на базе которого сделан ZX SPECTRUM. не попинает непосредственно слова БЕИСНКа, такие, как PRINT. IF. TAB. и т. д. вместо этого он выполняет приказы специального языка - своего внутреннего машинного кода, процедуры пзу. которые придают sPECTRUHy его индивидуальность, написаны на этой специальном языке и состоят из большого количества стандартных подпро

Табл. 1.1. некоторые примеры машинных команд Z80A.

Ссылка	Десятичное число	Мнемоника	Комментарий
(a)	61	LD D. c	загрузить в D содержимое с
<ы	14 27	LD c, 27	Поместить число 27 в с.
to	14 13	LD c. 13	поместить число 13 в с.
(d)	33 27 5e	ld hl, 13339	поместить 13339 в пэру регистров HL, Обратите внимание: 2V+a&6»?2M3339: 27 поместить в L: 52 по местить в н.
(e)	гг! зз гт 52	ld ix, 13339	Поместить 13339 в пару регистров IX.

водного отображения экрана в RAM <озу1 или копирования его обратно на экран с целы» создания эффекта мультипликации.

файл изображения и цветовые атрибуты занимают 6918 байт, следующая программа BASica сохранит отображение экрана, но это эаа-нет много времени - около то се-кунд:

5 CLEAR 58623 10 FOR 1=0 то 6911 во роке 58624*1,реек 116384*1) 30 кехт 1

Причина такой медленной работы в том, что SPECTRUM тратит больше всего времени на декодирование команд BASica перед их выполнением, некоторое количество времени также тратится на преобразование чисел в двухбайтную форму (которую понимает ZBOA) из десятичных чисел в пяти-байтной форме (с которыми оперирует BASIC). а также на выполнение пятибайтовой арифне тики..

так, в нашем примере по переброске экрана должны быть выполнены следующие шаги:

1. Прибавить 1 к 16364.
2. преобразовать результат в форму двух байтов.
3. Восстановить содержимое адреса реек.
4. Прибавить 1 к 58624.
5. преобразовать результат в форму 2-х байтов.
6. Сохранить полученное (реек) значение по заданному адресу

(роке).

т. Прибавить 1 к значение 1 и сохранить результат.

в. Вычесть 1 из 6911. Если результат положительный, то идти на пункт 1.

Во время прохождения цикла. SPECTRUM должен декодировать каждую команду снова, так как в данной случае память не используемся для сохранения последовательности предыдущих действий, легко увидеть, что компьютер тратит более 99 процентов времени на подготовку к выполнению задачи, а не на выполнение самой задачи. Аналогичная программа в машинных кодах для сохранения экрана выполняется практически мгновенно. Пример такой программы дан в Разделе в.

2. ВНУТРЕННЯЯ СТРУКТУРА ZX SPECTRUM

Компьютер - это машина, которая способна запомнить последовательность команд и затем вывод нить их. конечно, чтобы сделать так. требуется память, в которой команды могут быть сохранены. ZX 5PECTEUH имеет два типа памяти.

Первый тип - ПЗУ (10Н). в котором содержится фиксированная последовательность инструкций. введенных в машину ее изготовителей.

стартовый адрес или имя системной переменной	ячейка системной переменной	содержимое памяти
15384	__	Дисплейный Файл
22526	-	Атрибуты
23296	-	Буфер принтера
23552	-	Системные переменные
23734	-	карта никродрайва
CHAH5	23631	Область информации о каналах
PROG	23635	Адрес начала программы на БЕЙСИКе
VARS	23627	Адрес начала области программных переменных.
E- LI HE	23641	Адрес области редактирования
WORESP	23649	БУФСР 1ЖРШ-
STKBOT	23651	стек калькулятора
5TKEHЛ	23653	свободная область
SP	-	машинный стек и стек GO SUB
RAHTOF	23730	Пользовательские подпрограммы в машинном коде.
UDG	23675	Графика пользователя.
P-AHT	гзтзг	Физическая вершина ОЗУ.
Таблипа 2. 1. Карта памяти. Указатель стека SP хранится не в RAM. а в ЗР-регистре микропроцессора Z80A.		

ВТОРОЙ ТИП - ОЗУ (RAH). RAM -

это "блокнот для записей" компьютера. Когда компьютер выполняет задачу, он

непрерывно просматривает, что находится в RAH ("чтение" из памяти) и обновляет содержимое HAM ("запись" в память). SFECTRUH не использует свой "блокнот" для записей случайно. Различные части RAH используются для хранения различных видов информации. Программа BASica. введенная пользователем, например, хранится в одной части RAM, в то время как переменные, используемые этой программой, хранятся в другом месте. Размер "блокнота" для записей ограничен, и потому машина точна при распределении пространства для информации, которую она хранит, свободное пространство собрано в одном месте, и. если пользователь хочет добавить строку в свою программу, информация в RAM должна быть "перетасована" по всей длине, используя некоторую свободную область для вс тавки дополнительной строки.

в значительной степени эта глава посвящена объяснению организации RAM SFECTRUHa. так как многие программы раздела в предназначены для манипуляций с RAM. глава содержит в себе описание дисплейного файла, атрибутов, буфера принтера, системных перемен-вых. программной области и области программных переменных, в кон-пе раздела описываются стандартные подпрограммы из «ОН. к которым ссылаются программы в Разделе в.

Карта памяти

RAH имеет 49152 ячейки памяти. Каждая ячейка может хранить одиночное целое число от о до 255 включительно и задается адресом.

КОТОРЫЙ является положительным целым числом от о до 65535. Адреса от о до 16383 зафиксированы для постоянной памяти - ROM. первый адрес RA» (ОЗУ) - 16384.

Табл. 2. 1. - упрошенная карта памяти sPECTRUHa, которая показывает, как используется RAM с адреса

1БЗВ4.

Дисплейный файл. который хранит отображаемую на экране информацию, занимает ячейки от

16384 ДО 22527. атрибуты, которыв определяют цвет. яркость я т. д. для каждого знакоместа экрана, следуют непосредственно дальше:

В ячейках 22528 - 23295.

Первые 5 адресов в колонке 1 Таблипы ?.. 1. являются фиксированными, т. к. дисплейный файл, атрибуты и т.д. занимают фиксированное положение, пятая область назначена для карты памяти НИКРО-драйва. это небольшое периферийное устройство представляет собой вечто среднее между магнитофоном и дисководом. Грубо говоря - это дешевая альтернатива дисководу. скорость его работы достигается за счет того, что .с одной стороны, дрнта движется с очень высокой скоростью, а с другой - за счет организации прямого доступа к файлам, как на диске, а не последовательного, как на магнитофонной кассете, вот для того. чтобы иметь в некоем месте хранилище с данными о том. что записано на каждом секторе микродрайва, в памяти компьютера и выделена область карты памяти иикродрайва. Если микродрайв (а точнее INTER-FACE -1. через который он подключается), подсоединен к SPECTRUMу. эта область содержит информацию о его секторах. Если же не подсоединен, эта область не нужна и

в этой случае шестая область (информация о каналах) размешена непосредственно за четвертой областью, системными переменными. т. е. стартовый адрес области ян-Формации о каналах и всех исследующих областей не фиксирован, а может "плавать" вверх-вниз в RAM-

SPECTRUM хранит стартовый адрес всея этих областей в системных переменных. Область системных перемешшк находится перед картой кикродрайва в ячейках 23552 - 23733 включительно, эти адреса являются все время строго фиксированньвш.

Адреса ячеек, которые хранят стартовые адреса всех 'плавающих" областей, дана в колонке 2 таблицы 2. 1. Адрес области начала программы BASICa, например, хранится в ячейках г3635 и гзбЗб в области системных переменных.

Примечание: Ссылка к системной переменной с домошью адреса. DO которому она хранится, довольно неудобна. По этой причине в Таблице 2. I. в 1-ой колонке даны условные

имена системных переменных, эти имена удобны только для пользователей, в то время как SPECTRUMOH они, естественно, не распознаются, например, введенная строка:

FRINT PROG

даст сообщение об ошибке •г: variable not found" ["Переменная не найдена").

FEES и POKE

Карта памяти - это ключ для понимания того, как компьютеров используется RAH-панять. Для непосредственного управления RAH используются ключевые слова BASica - PEEK и POKE, которые ао-зволяют просмотреть и изменить содержимое любой ячейки памяти.

PEEK - функция вида: PEEK адрес

Адрес - это целое положительное число ОТ 0 ДО 65535 или зрифнетическое выражение. которое. выполнят: ь, дает положительное число. Важно заключить арифметическое выражение в скобки, т. к,

PEEK 16364 + г

интерпретируете я. как

(2 * результат PEEK 16384),

тогда как

PEEE (16364 + 2) интерпретируется, как

PEEK 15366

значение, выдаваемое функцией PEEK - это число, хранящееся по данному адресу в текущий момент. Оно всегда будет положительным ОТ 0 ДО 255.

Выше об'яснялось, что системная переменная P8OG хранится во адресу 23635, но значение FROG (т. е. адрес в КАШ всегда больше числа 255. Следовательно, для его хранения необходимы две снежных ячейки с адресами 23635 и гзбЗб. Значение PROG может быть подучено с помощью командной строки:

PRINT -PROG=": PEEK 23635 + 256"PE?K гзвЗб

программа рг.1

```
100 FOE 1=0 TO 703
110 PRINT " -; : Rai символ, находящийся на клавише "5- в G-рехиме
120 KEXT 1
130 SAVE "Picture" SCREEH*
140 CLS
150 INPUT 'Перемотайте ленту и включите воспроизведение"; т»
160 LOAD "Picture- SCREEH$
```

все адреса хранятся в г-х смежных ячейках в такой форме и могут быть подучены вводом: PRINT PEEK 1-я ячейка * 256«PEEK 2-я ячейка

например, если SPECTRUM, используется без подсоединенного никродрайва. область карты МИКРО-драйва не будет существовать, и информация о каналах будет располагаться непосредственно после области системных переменных, т. о системная переменная CHAHS должна быть такой же. как стартовый адрес карты никродрайва. когда он существует, т. е, 23Т34. CHAHS хранится в 23631 и 23632 и, следовательно, после ввода:

PRINT PEEK 23631 * 25Б-PEEK 23632

будет получено значение 23734.

функция PEEK может быть использована также для просмотра содержимого любой из ячеек ПЗУ. Это очень полезно. Просмотр любой ячейки не приводит к разрушению или искажению программы или переменных. иногда результаты PEEK могут быть обманчивыми, т. к. содержимое ячейки, которая просматривалась, может изменяться в течение иди непосредственно после выполнения команды просмотра.

Например, если просматривается содержимое ячеек, которые связаны с левым верхним углом экрана дисплея, то результаты, распечатанные в верхнем левом углу экрана. будут уже устаревшими, т. к. в момент вывода изображения мы уже изменили эти ячейки.

Команда POKE более рискованная, чем функция PEEK. т. к.. задавая ее. пользователь вмешивается в функционирование компьютера. Использование этой команды может быть

причиной сбоя машины или ее останова.

Формат команды: POKE адрес, чнсло

Адрес - это положительное целое число ОТ 0 ДО 65535 включитедьно или арифметическое выражение, которое дает такое число, в этом случае нет необходимости заключать арифметическое выражение в скобки, т. к. POKE - это команда, а не ФУНЮШЯ {хотя есть негласная заповедь, что лишними скобками программу не испортишь). Число* помещаемое в ячейку, может быть от 0 ДО 255 вкдючительно.

Дисплейный файл

Обычно дисплей содержит 24 строки по 32 символа. Дисплейный файл занимает ячейки от 16364 до 22527. т. е. 6144 ячейки в обшей сложности, /следовательно, количество ячеек, используемое для символа:

6144/(24"32)=8.

Эти 8 ячеек формируют изображений па символа экране, называемое знакоместом.

Наиболее легкий путь получения обвего впечатления о тон. как организован дисплейный файл - это печать <PE1HT) картинки на экране, сброс (SAVE) экрана на ленту. очистка экрана (CLS) и загрузка (LOAD) картинки экрана вновь, программа P2. 1 сохраняет (SAVE) и загружает (LOAD) экран, используя графический символ на клавише 5 для создания оригинальной картин-ки,

Когда картинка загружается с ленты, видно, что дисплей разделен на три зоны по в символьных строк в каждой:, а каждая строка разделяется на восемь пиксельных линий. SFECTRUK загружает сначала верхние пиксельные линии для первых восьми строк, затем следующие пяксельвые линии тек же восьми строк и т. д. таким же образом формируются средняя и нижняя части дисплея.

Другой путь понимания формирования дисплея - это рассмотреть, где хранятся в байтов, которые используются для формирования символа в верхнем левом углу экрана, первый байт Формирует самую верхнюю 1/8 часть символа и размешается в начале дисплейного Файла по адресу 16384.

Команда POKE 16384,0 очистит верхнюю линию пикселей самого верхнего левого знакоместа, в то время, как РОЖЕ 16384, ?55 закрасит всю ЭТУ линию. при помещении в эту ячейку числа от о до 255 мы получим в этом месте экрана различные штрихи, вторая сверху линия в первом знакоместе на экране не сформирована числом, хранящимся в ячейке 16385, - эта ячейка используется для верхней линии пикселей в соседнем символе. Вторая линия сверку в первом знакоместе формируется числон. хранящимся в ячейке

16384*32»8= 16640.

Подобным же образом вычисляют^ ся адреса оставшихся шести линий этого знакоместа.

Следовательно, образ символа в верхнем левом углу экрана определяется содержимым адресов 16384,

1664О, 16896. 17152. 17406, 17664, 17920. 18*76.

программа P2. г. позволяет вам экспериментировать, помещая различные чисаа в эти восемь ячеек. Каждая* ячейка в дисплейном файле определяет восемь пикселей на экране, при этом число, которое хранится в данной ячейке, преобразуется в двоичную- форму и затем устанавливаются восемь пикселей, соответственно двоичным цифрам. Например. 24О после преобразования в двоичное число дает: П1i0000

следовательно, если ячейка со• держит число 240, четыре из восьми пикселей, соответствующие единицам, будут высвечены, а оставшиеся -- нет.

следовательно, дисплейный файл состоит из 6144 ячеек памяти по 8 ячеек для одного знакоместа. Каждая ячейка определяет горизонтальную полосу из восьми пикселей, ячейки, относяяиеся к данному знакоместу, не расположены последовательно одна за другой.

Атрибуты

Содержимое дисплейного файла определяет, какие пиксели высвечиваются на экране. цвет фона (PAPER). символа (INK). яркость (BRIGHT) и мерцание (FLASH) определяются с помощью атрибутов. Область атрибутов занимает ячейки

22528 - 23295 - ПО ОДНОЙ ЯЧ6Йке для каждого из 768 знакомест.

соответствие между содержиным ячеек памяти файла атрибутов и самими атрибутами - следующее:

значение атрибута 128«FLASH + 64*BRIGHT » fl«PAPER + INK

FLASH И BRIGHT ПРИНИИЭЮТ ЗНЭчение i. если соответствующее условие установлено, а PAPER и 1НК принимают значение требуемо го авета. как показано на клавиа туре (красный, например,- 2).

Программа РВ.3. декодирует атрибуты, т. е. данное значение атрибута распечатывает с соот ветствуюшнн цветом PAPER и IKK.

Буфер принтера

256 ячеек ОЗУ. следующие за областью атрибутов. используются. как буфер для хранения строки символов, которая должна быть пе редана на принтер.

Многие программы в Разделе В будут использовать буфер принте ра для передачи данных BASICa или от клавиатуры в подпрограммы. буфер подходит для этой цели. т. к. его ячейки фиксированы и малове роятно, что пользователь пожелает его использовать для ДРУГИК целей.

Единственно важное ограничение з этом случае не использовать команды BASica. которые треехют работы с принтером - LLIST. LPRINT. COPY.

Есть еще и ограничение для владельцев 126 килобайтных наяин. у них нет буфера принтера. Дело в том, что буфер принтера предназ начался в оригинальной модели для поддержки недорогого спеииализированного узкопечатного zx принтера. В фирменной модели 128 кияобаитных машин есть порт подключения полноценного матричного принтера, обладающего собственным буфером и необходимость в этом буфере отпала, зато в этих моделях необходимо больше системных переменных и под них отдали область буфера zx-принтера. теперь, если в режиме 126E пользователь что-либо зашлет в эту область, то нарушив системные переменные он выведет программу из строя.

Программа рг. г. программа для создания символа в верхнем левом углу экрана.

```
ю ЙЕН подпрограмма установки символа в верхнем углу экрана
20 INPUT "Символ состоит из восьми байтов, каждый из которых - число в диапазоне от о до
      255. введите номер байта (от 0 до 7)":п
30 IF IKO OR П>7 OR n<>INT П THEH BEEP 0.2, 24. GO TO 20
40 INPUT "Ввести содержимое байта";m
ЬО IF Ш<0 OR Ш>255 OR molHT Ш THEK BEEP 0. 2, 24: GO TO 40 60 POKE 16384+8*32*n.m
программа Р2.з. программа декодирования атрибута.
10 REM декодер атрибутов
20 DATA "BiacK", "Blue", "Red". "Haeenta". "Green", cyan".
"Yellow". "White". "Bright". "Flash-30 DIM C*(8,7) 40 FOR 1=1 TO в 50 READ C*ll) 60 NEXT
      1
100 REM Декодер атрибутов
но IRPUT "Ввести число от о до 255. эта программа декодирования интерпретирует его в
      файл атрибутов";п 120 IF П<0 OR П>255 OR П<>1ЫТ П THEH BEEP ,2,24:60 TO ПО 200
      PRINT "Цвет символа "; C*(l+n-8»INT(n/8))
гю PRINT "Цвет Фона - "; с»(i*iaTin/8)-8*INT(n/64)) 220 IF INT (п/б4)м OR INT ш/б4) = з
THEH PRINT "СИМВОЛ - BRIGHT"
гзо IF п>127 THEH PRINT "символ будет мешать (FLASH)" зоо PRINT AT б. о;
      ":::::::::::::i:::::::::::::::::;•
310 FOR 1=22720 TO 22751 320 POKE 1.П 330 NEXT 1
500 INPUT "Для повторения - ENTER"; z*
510 CLS
520 GO TO 110
```

Область программ на BASIC

6 Обычно эта область начинается с адреса 23755 и на нее указывет содержимое системной переменной PROG (23Б35,23636). -но есть и ис-ключения за счет некотрых видов периферийных устройств,

Если, например, к компьютеру подсоединен Никродрайв, то начало этой области сдвигается, в этом самом обшей случае начало области программ на ВАЗICe и определяют с пометь» системной переменной PROG. Ниже предполагается, что такая периферия не подсоединена.

Программа Р2.4. распечатывает содержимое 18 ти ячеек в начале программной области, как показано на рис. 2.1. в этих 18 ячейках хранится первая снрока данной программы.

```
ю KEM Peек program
20 FOR 1=23755 TO 23772
30 PRINT 1.PEEK 1
0 NEXT 1
```

Программа Р2. 4,

Программа для просмотра содержимого первых 16-ти ячеек в программной области.

```
23755 О
23756 10
E3757 14
23758 О
23759 Й34
23760 вО
23761 101
23762 101
23763 107
23764 32
23765 112
23766 114
23767 111
2376S 103
23769 114
23770 97
23771 109
23772 13
```

Рис. г. 1. Форма, в которой строка ю REM peek program хранится & программной области.

Номер строки (Ю) хранится в первых двух ячейках в форме:

Номер строки - 256«PEEK первый адрес + PEEK 2-й адрес.

Следуюшие 2 ячейки: 23757 и 23756 хранят длину оставшейся части строки, начинающейся в ячейке 23759. в нашем ел/чае:

14+256*0=14

Т.о. следующая строка начина ется с ячейки:

23759*14 23773

Ячейка E3759 хранит в себе чи ело 234, которое является кодом ключевого слова (токена) KEH. Следующие 12 ячеек хранят коды символов одиннадцати букв и ПРО -бела, составляющих фраз/ PeeK

program. Последняя ячейка хранит число 13. которое является кодом для EHTEF. определяющий конец строки.

Таблица 2.2. показывает нетод кодирования программы в программной области.

Ячейки	содержимое
1 и г 3 И 4 5 конец	номер строки длина строки, исключая первые 4 ячейки код команды символ OTTER, число 13
Табл. 2. 2. Этот нетод используется для кодирования строк программы.	

Нонент. который опушен в таблице - это описание метода хранения числовых значение, имеювжх место в программе. Этот метод может быть исследован на примере строки:

10 LET a^1443

Введите ее в программу РЕ.ч. на РИС. z. Z показан результат работы программы в

этом случае.

```
23755 0
23756 10
23757 14
23758 0
23759 241
23760 97
23761 61
23762 49
23763 52
23764 52
23765 51
23766 S4
23767 0
23768 0
23769 163
23770 5
23771 0
2377г 13
```

Рис. г. 2. Формат, в котором строка ю LET а-1443 хранится в программкой области.

Ячейки 23755-2375в такие же. как в прошлом примере, затем следуют коды для LET. а. - и четыре кода цифр, которые вместе формируют число 1443. Следующий элемент, находящийся в ячейке

23766. - это код 14. Этот код указывает, что следующие 5 ячеек хранят число в специальном пяти байтном формате. Ливия заканчивается в ячейке 23772 вводом символа ЕЫТЕЕ. как и ранее.

Примечание ИНФОРКОМа.

Получается так, что в одной строке число как бы записано дважды, первый раз своими символами. а второй раз - в пятибайтной форме после кода 14. Зачем это нужно?

Дело в тон. что первое представление используется для того. чтобы БЕИСИй-интерпретатор знал. что вам показать на экране, а

Ю PRJHT "Ввести экспоненту и четыре байта мантиссы.

все числа должны находиться между о и 255

включительно. " 20 IHFUT e, a, Ь. c.d зо PRINT .. " Exponent- ":e 40 PRINT -Mantissa; ":a.,b,.c..d so PRINT .. -The number^ -;(z»ta<i26)-i)«2A(e-i60>*

(< (256« (a+126* (a<i28»+b) «256*О "256+d)

Программа РЕ. 5, эта программа восстанавливает дробное число.

Тип переменной	диапазон символьногокода	длина в области переменных
цифровой (односимвольное имя!	97 - 122	б
цифровой (многосимвольное имя!цифровой массив	161 ~ 186 129 - 154	5 + длина имени ч- + 2*разнерность + 5»обшее количество элементов.
Управляющая переменная цикла FOR - NEXT	225 - 250	16
символьная строка	65 - 90	3 + дляаа строки
символьный массив	193 - 216	4 + 2«разиерность + обшее число элементов.

Таблица 2. 3. Переменные. Диапазон кодов и длина переменных.

второе - используется для расчетов во внутреннем калькуляторе компьютера. (О калькуляторе читайте в т. 1 и 2 нашего трехтомника).

самое интересное, что они могут и не совпадать, в этом случае Вы на экране будете видеть одно. а программа будет обрабатывать совсем другое число, и этим нередко ПОЛЬЗУЕТСЯ з защите программ, например, вы видите на экране LET а^ 1257: GO ТО а И

пытаетесь проследить работу программы, а на самом деле таи было записано нечто совсем другое и компьютер делает переход не к строке 1257. а туда, куда надо.

Желающие могут посмотреть загрузчик программы BOHBJACK. в череде относительно несложных вывертов этот там стоит одним из первых, но если на него 'КЛ10НУТЬ', атака на программу никак не подучится.

Цифровой пятибайтныи формат

Пять ячеек памяти используются для хранения чисел в программе на BASice (исключая номера строк). целые числа в диапазоне от -65535.до *65535 хранятся та-ким же образом, как в формате Z80A, для этих чисел первые две ячейки и последняя содержат О. а третья и четвертая хранят число в

двухбайтной Форме:

число = PEEK 3-я ячейка + 25б«PEEE четвертая ячейка таким образом 16533 хранится в пяти ячейках как

о о 169 64 о

ПОТОМУ. ЧТО

169*2Ъб«64-1б553

Дробные числа хранятся в форнате с плававшей запятой таким образом: экспонента в первой ячейке, а мантисса в следующих четырех ячейках, т. е. : число - мантисса * 2*экспонента

Первая ячейка мантиссы используется также для определения знака числа. ЕСЛИ ячейка содержит значение в пределах от о до 127. число положительное, если нет отрицательное.

Программа Р2. Ь может быть использована для восстановления дробного числа из составляющих его зяти компонентов.

Область переменных

Область переменных начинается в ячейке, адрес которой хранится в системной переменной VARS (ячейка 23627). как бы ни бьыа объявлена новая переменная, т.е. в программе или непосредственным вводом с клавиатуры, для нее резервируется соответствующее количество свободного пространства в этой области.

Все имена переменных начинаются с бгквы. Различий между верхним и нижним регистром нет. Эти ограничения позволяют ЗРЕСТКТЖУ манипулировать с кодом первого символа каждой переменной и. таким образом, он может различить шесть типов переменных просмотром диапазона, в которой находится код.

Все цифровые переменные с односимвольными именами. например, имеют коды в пределах от 97 до \гг; буква а - 97; ь - 98; с - 99 и т. д. подобным же образом ПИФРО вые массивы имеют коды в пределах 129 - 153, т. е- а - 139; b - 130; с 131 и т. д. Диапазоны кодов представлены в таблице Z. 3. Длина каждого типа переменной также показана в таблице 2. 3.

Подпрограммы ПЗУ

Некоторые из представленных в книге программ (раздел В) используют стандартные подпрограммы ПЗУ:

RST 16 - распечатывает содержимое аккумулятора.

CALL 3976 - вставляет символ. хранящийся в аккумуляторе, по адресу, хранящемуся в паре регистров HL,

CALL бЗЕб - если аккумулятор хранит код 14. устанавливаете я нулевой флаг и увеличивается пара регистров HL в пять раз.

CALL 6510 • возврат в HL адреса в ОЗУ той строки, номер которой был передан в ЭТУ подпрограмму через HL,

3. МАШИННЫЙ язык Z80

Регистры Z80A

Во время выполнения программы компьютер не обновляет непосредственно содержимое памяти, он копирует содержимое ячейки памяти в регистр и оперирует содержимым регистра. Регистры в машинном языке имеют функцию, схожую с переменными в BASice. т.е. используются для хранения чисел, они отличаются от переменных BASICS тек. что число их ограничено и они существуют в процессоре, а не в КАИ. Кроне того, они могут хра нить только один байт, или г байта в паре регистров.

Z80A - инеет несколько регистров и потону может хранить несколько чисел одновременно. Благо даря этому снижается время при обмене информацией между цроцес сором и памятью.

Регистр "А" (Аккумулятор)

АККУМУЛЯТОР - это наиболее важный регистр, т. к. больше всего арифметических и логических команд выполняется с содержимым этого регистра. Этот регистр называется аккумулятором, т. к. результат последовательных операций накапливается ("аккумулируется-) в нем. некоторые команды, которые

Флаг	мнемоника		Ис пользование
знак	«	P	включается, когда результат последней операции отрицательный.
нуль	Z	NZ	включается, когда результат последней операции равен НУЛЮ или имело место совпадение.
перенос	C	NC	Включается, когда происходит переполнение регистра, т. е. последний результат больше тсго. что иожет быть записанным в один байт (или в два байта для операций с парой регистров).
Четность/ Переполне ние	PE	PO	Флаг включается, если в байте результата предыдущей операции количество включенных битов есть величина четная, в некоторых операциях этот флаг свидетельствует о переполнении.

таблица 3. г. четыре флага, которые контролируют наибольшее количество операций Z80A.

обращаются к аккумулятору, используют другой регистр или адрес памяти в качестве источника данных, например. инструкция

ADD A, B

дает процессору команду прибавить содержимое регистра B к содержимому регистра A. поместив результат в A.

Флаг-Регистр -F" (Регистр Состоянии)

Регистр F довольно значительно отличается от всех остальных, т. к. его содержимое не рассматривают, как один байт, а рассматривают как 8 индивидуальных битов, что конечно же одно и то же. эти биты используются в качестве так называеих флагов для управления последовательностью выполнения программы, каждый флаг используется для определения того, какое из двух логически противоположных условий имело место при выполнении предыдущей операции, Например, флаг нуля определяет, был ли равен нулю результат последней операции сложения, вычитания и т. п. Только 4 из восьми флагов наиболее интересны для пользователей, их свойства кратко изложены в Таблице з. г.

Флаг знака наиболее простой, кроне того, условлено. что в one-рапняк со знаковыми числами седьмой бит (старший бит в байте) тоже используется для хранения знака, он включается, когда число отрипательно. иначе снимается.

Этот флаг отражает знак последнего результата.

Флаг нудя устанавливается, ес-

равен нулю, он также используется инструкциями сравнения, которые фактически аналогичны вычитанию, при котором результат игнорируется.

Флаг переноса регистрирует переполнение, которое имеет место, если результат сложения длиннее чем то. что может быть записано в регистр, либо имеет место заем

при вычитании, имеется также несколько инструкций сдвига, в которых биты в регистре А сдвигаются влево или вправе циклически через флаг переноса.

Флаг четности/переполнения это два флага в одном. Он используется, как флаг переполнения при выполнении арифметических операции для определения, был ли бит 7 получен в результате переноса или сгенерирован битой б при "взятии займа", это используется в случае, если бит знака был искажен. Логические инструкции используют тот же самый флаг для определения четности результата.

Примечание: четность двоичного числа это четность количества битов, установленных в единицу. Если количество четно, то говорят, что имеет место четность. если оно нечетно , то говорят, что имеет место нечетность. Флаг устанавливается, если имеет место четность.

Результат некоторых инструкций зависит от текущих установок отдельных Флагов, например, инструкция

Jfi Z. D

позволяет Z8GA "перепрыгнуть" через несколько инструкций, если флаг нуяя установлен (т. е. сделать условный переход типа IF. . . тнек GC то). Если флаг нуля не установлен, процессор выполняет следующую инструкцию в обычной последовательности, т.о. Флаго-вый регистр очень важен, т. к. он! позволяет процессору принимать! решения и переходить к другой части программы.

Регистры Счетчики "B" и "C"

Регистр в и, в некоторой степени, регистр С. с которым он может использоваться в паре, доступен для нескольких применений, но наиболее важно использование его в качестве счетчика. Ны уже видели, как выполнение программы но лет управляться использованием Флага о в инструкции JR z, n.

1

другая инструкция: DJNZ n исаользует флаг о для создания иикло в, используя регистр в в качестве счетчика аналогично циклам FOR-NEXT в BASICS. Когда встречается эта инструкция, ZBOA уменьшает содержимое регистра В на 1. Если результат равен нулю, то выполняется следующая инструкция в нормальной последовательности. Если результат не равен нулю. подпрограмма "перескакивает" n инструкций. Если программист использует отрицательное значение для n, "прыжок" осуществляется в программе назад и. если здесь нет других переходов, процессор. е конце концов, встретит ТУ же самую инструкцию вновь.

Т. о. загрузкой регистра в определенным числом и установкой соответствующего смешения n определенная часть кодов может быть выполнена заданное количество раз. Регистр в содержит только один байт и поэтому ножет устанавливаться на любое число от о до 255. т. е. используя описанный механизм через определенный блок кодов может быть сделано максимум 255 проходов, для осуществления более 255 проходов в цикле нет эквивалентных инструкций, но имеются очень мощные инструкции, которые используют все 16 битов регистровой пары вс, как счетчик с максимальным значении 60535. например. янструкиия CPDR. КОГАЗ

Z80A выполняет ее, то:

П содержимое ВС уменьшается на 1;

2) уменьшается содержимое HL (о регистровой паре HL см. ниже);

3) сравнивается содержимое аккумулятора А с содержимым ячейки памяти, адрес которой находится в паре регистров I1L.

Процессор повторяет эти действия до тех ПОР. пока не будет совпадения между содержимым А и ячейки паняти, либо сока не обну-лится ВС, Т- о. эта инструкция может быть использована для поиска ячейки, содержащей определенное число.

Адресные регистры "DE" и "HL"

Регистры D и Е не имеют особых функций И В основном используются в качестве време иной быстродостушюй памяти, они также используются в паре для хранения адреса ячейки, которая интересует нас в текущий момент.

Основная функция регистров и и L - хранить в паре адрес ячейки памяти. Ни уже видели, как определенная инструкция использует пару HL с этой целью, и кранит старили

байт, L - младший, полный адрес доступен в Форме:

адрес - 25&«н + L, что дает максимум 65536 возможных адресов.

Индексные регистры "IX" и "IY"

Индексные регистры IX и IY являются 16-битовыми регистрами и могут быть использованы только так в отличие от регистров вс. DE и HL, которые могут использоваться и в паре, как 16-битовые регистры, и индивидуально. как в-битовые. IX и IV. главным образом, используются подобно паре HL.

IX и IV, кроне того, имеют одно свойство, которое недоступно для HL - они могут быть использованы со смешением з. имеется в виду, что инструкция. которая ссылается к UX+s), использует не ячейку памяти, адрес которой хранится в IX. а сначала смешение з прибавляется к значению в IX. чтобы получить новый адрес, и с ним и работает ИНСТРУКЦИЯ.

Указатель стека "SF"

стек это область в верхней части ОЗУ, которая используется для временного хранения содер жимого пар регистров, стек рас тет вниз, когда идет заполнение и поднимается вверх при его опустошении, начало стекл фиксировано В ZX SPECTRUM - оно нэходится непосредственно ниже ячейки, на которую указывает системная переменная RAMTOP. Вершина стека находится ниже нижней границы стека, т. к. стек увеличивается вниз, а уменьшается вверх. Адрес текущей ячейки верхней границы стека хранится в SP-регистре, передача в стек или из стека происходит с помошью инструкций

PUSH и тор. например:

PUSH HL в этом случае процессор;

1) уменьшает SP:

2) кодирует содержимое регистра и в ячейку, указанную указателем стека SP;

3) уменьшает SP;

чз копирует содержимое регистра L в ячейку, /казанную указателем стека SF.

Инструкция POP выполняет противоиоложное действие, по этому методу последняя поступившая пара чисел, помешенная в стек, является всегда первой парой, которая снимается со стека, это обесвечи вает простой и удобный с пособ временного хранения содержимого регистров во время вызова подпрограмм.

Программный счетчик "PC"

Программный счетчик PC - очень важный 16-битовый регистр, т. к. он хранит адрес следующей инструкции, которая должна быть выполнена, нормальный код работы -когда инструкция выполняется следующим образом:

1) Скопировать содержимое ячейки. указанной программным счетчиком PC в специальный регистр процессора.

2) Если инструкция содержится в нескольких байтах, увеличить PC и скопировать содержимое сле-дуюшей ячейки во второй специальный регистр.

3> Увеличить PC таким образом, чтобы он указывал на следуюяую инструкцию, которая должна быть выполнена.

4) выполнить инструкцию, которая только что была прочитана.

Команда JUMP так же. как и DJNZ n или JR 7.. п. изменяет нормальный ход ьыполнения программы с помощью изменения PC во время выполнения шага t. заметим, что это изменение ииеет место после увеличения PC. таким образом, значение смешения n всегда будет отсчитываться относительно позиции инструкции, следующей за инструкцией, содержащей смешение.

Альтернативные регистры AF', BC', UK', HL' !I

Z60A имеет копию каждого из! A. в. с. D. E.H и L регистров. Отли-! чаются копии использованием зна-1 ка " ' " возле обозначение Ре-1 гистра. например. A' - это коиия! регистра A. инструкции не работа 1 ют с этими регистрами непосредственно, но инструкции обмена по эволяют вывести из использования 2 или более регистров и ввести в использование вместо них их ко пни.

Команды обмена выполняетея очень быстро, т. к, содержимое регистра физически не

копируется из одного регистра в другой, а вне сто этого установка внутренних коммуникаций процессора изне-няется так, что дублирующий ре ГИСТР становится основным, а ори гинальный регистр - запасным.

О системе команд процессора Z80

В системе команд Z80 около 700 инструкций. Поскольку возможны только 256 различных комбинаций из 8 битов (г"в 256 1, ТО лишь менее половины команд может быть выражено одним байтом, оставшиеся инструкции хранятся в двух или трех байтах. некото!«ые команды следуют с однобайтовым смешением з. или однобайтовын числом п. или двухбайтовым числом <адреСОН) ПП. К которым инструкция ция может занять максимум ч байта.

ПРИМЕЧАНИЕ "ИНФОРКОМа"

Zx SPECTRUM ра с пола гает еще встроенным программируемым калькулятором. ИМРКЮИН свою систему команд. Доступ к ним возможен только из машинного кода, конаыды калькулятора могут быть более длинными, чек команды процессора и занимать более 1-х байт. (Сн. т. 1 нашего трехтомника).

DIM

для удобства работы на ассенб дере все команды записываются с помошью мнемоник (OF CODE), нне-ноника - это сокращенное описание каждой команды.

Правила задания команд в ассемблере:

1. одиночные регистры описываются буквой, например - в. Пара регистров именуется в алфавитном порядке, например, вс.

а. смешение з положительно, если оно наводится в пределах от о до 127. и отрицательно, если оно находится в пределам от las до 255. Большие или меньшие числа недопустимы.

Отрицательное значение подсчитывается вычитанием з из ?56. например, команда относительного безусловного

перехода:

JR 3

вызывает переход вперед на в байтов, если з-8 и переход назад на в байтов, если s-248, т.к.

гъ&-е^г*в.

запомните при подсчете смешения, что иерекод осуществляется с адреса первого байта следумней команды.

3. однобайтовое число п лежит в пределах от о до 2Ь5 включительно.

4. Двухбайтовое число или адрес представляется как пп и лежит в пределах от 0 до &Ъ?>35 включительно.

5. гш. заключенное в скобки. т. е. (гш). подразумевает "содержимое ячейки по адресу пгг, тогда как пп без скобок подразумевает "число пп".

Т.О. . LD HL, (23&27)

подразумевает загрузку пары регистров HL содержимым ячеек азьгт и 23&28, тогда как: LD HL, ЙЗЬ^У подразумевает загрузку HL числом

J3627.

Таким же образом. (НЫ подразумевает "содержимое ячейки с адресом, хранящимся в HL•. тогда как HI. без скобок означает "число, хранящееся в HL".

Такой вид адресации называется косвенной адресацией.

о. место назначения результата операции всегда задается первым. Например:

ADD A. B

означает "прибавить содержимое регистра В к содержимому регистра А и результат оставить в регистре А".

РАЗДЕЛ В

4. ВВЕДЕНИЕ

В Разделе в представлены программы в машинном коде, для простоты и удобства использования они даны в стандартном формате, во введении описывается этот формат и дается программа на BASice, которая может быть использована для загрузки программ в память.

длина: это длина программы в байтах.

Количество выполнение некоторых переменных: программ может потребовать изменения значения одной или более переменных, передаваемых в программу через буфер принтера.

Контрольная каждая программа да-суниа: па в виде последовательности целых положительных чисел, помещаемых в последо вательные ячейки памяти, контрольная сунна (т. е. сумма всех чисел вводимой подпрограммы) дается для того, чтобы Вы были уверены в DPS вилькости загрузки. функция: дается краткое описание задачи, выполняемой с ионошью программы. Неременные: определяются имя.

длина и адрес в буфере принтера каждой переменной, переменная длиной в один о'айт (целое положительное число в ире делах от 0 до 2551 передается в программу из BASica или с клавиатуры через

POKE.

POKE ячейка, значение Двухбайтовая переменная передается с помощью г-х команд: POCE ячейка, значение

- гьб«1HT(значеняе/г5б): POEE ячейка+1.1HT(значение/25б) ! используемые ячейки являются I ячейками памяти буфера принтера

вызов программы вызываются подпрограммы: с использованием

ФУНКПИЮ USR. которая должна быть включена в команду. ЕСЛИ ПРО грамма в машинном коде не передает ка кое-то значение обратно а BASIC по завершении, то используется команда:

KANDOHIZE US» адрес

Если Вам надо, чтобы результат был возвращен в регистровой паре BC, то вызов делается так:

LET A = US8 адрес иди

PRINT USR адрес в зависимости от того, должны ли возвращаемые данные сохраняться в переменной BASICa или выводиться на экран.

КОНТРОЛЬ Объясняются оровер-ошибок: ки. выполняемые

программой для нелогичных или противоречивых значений переменных, параметров и т. п.

Комментарии: Объясняются возможные варианты в программах.

листинг в программы представ-машинном лены на языке ассем-коде: блера. для загрузки в память используется третья колонка - "Числа для ввода", все числа здесь даны в десятиричной системе.

как она Объяснение принципа работает: работы программы.

Загрузчик машинного кода

Почти все программы из этой книги перемещаемые, т. е. они будут работать корректно независимо от того, в каком месте КАИ мы их поместим. Если программа не перемещаемая, то в комментариях объясняется, как она должна быть изменена, если ее нужно сохранить в другой области памяти, в Разделе А <часть 2) мы видели, что SPECTRUM использует различные части RAH для различных функций и что область между ячейками, указанными системными переменными RAMTOP и UDG, предназначена для хранения подпрограмм в машинном коде.

Программа BP может быть использовала для загрузки, изменения и перемеиения программы в машинном коде, с ее неновые* пользователь может переустановить указатель RAMTOP, что даст больше свободного пространства для на шинных кодов; ввести

программу с клавиатуры; перейти вперед или назад для корректировки ошибки: вставить или удалить часть про граммы:

Когда программа BP запускает ся. она печатает младший адрес, с которого программа в машинных кодах может быть введена и сохранена, т.е. на единицу больший, чей RAMTOP.

в машине с чек памяти младший

адрес - &536B, если пользователь не обновлял системную переменную

RAMTOP.

В конце ОЗУ обычно резервируются 168 байтов для UDG. но программа позволяет пользователю использовать и эту область, если ои пожелает, он может также выбрать новый возможный младший адрес, который программа затем помешает в системную переменную RAMTOP, используя команду CLEAR, данные не могут быть введены но адресу, меньшему, чем 2700О. т. к. иначе нарушатся границы области, тре-буюшейся для самой программы BP. Программа BP запрашивает адрес, с которого должна стартовать ПРО грамма в машинных кодах, т. о. пользователь может резервировать область для нескольких процедур и затем загружать их каждую отдельно.

РИСУНОК BF1 показывает формат дисплея после того, как с ячейки 32000 была загружена программа "Screen invert", первая колонка -адрес, вторая - содержимое ячейки памяти с этим адресом, третья -контрольная сумна. программа "Screen invert" - инеет 18 байтов в ДЛИНУ и ее контрольная сумма -1613. следовательно, она занимает ячейки от 32000 до 32017. и контрольная сумма дана для ячейки зго!7, т. е, сумма содержимого ячеек (Зг000... зг017) равна 1613.

на основном экране вникание пользователя привлекается к одной ячейке - содержимое этой ячейки мерцает. Эта ячейка является те-кушей и первоначально это выбранный стартовый адрес программы. Пользователь вводит целое число между о и 255 включительно, ко торое программа HC LOADER помешает в текучую ячейку, затем следующий адрес становится текуией ячейкой.

Пользователь может не вводить число, а вместо этого выдрать для корректировки вариант, описавши в таблице BTI.

код	вариант
в	Перейти иа один адрес назад.
B п(число)	перейти на n адресов назад.
F	перейти на один ад рее вое ре д.
F n (ЧИСЛО)	Перейти на n адресов вперед.
I n (ЧИСЛО)	вставить n байтов, каждый из которых содержит 0.
D n C ЧИСЛО)	Удалить n байтов в текущей области.
т	закончить программу

Таблица BTI. возможные варианты редактирования машинного кода.

Программа BP. загрузчик малинного кода.

(HC LOADER)

```
100 GO SUB и100
J00 KEM »**«« вычисление ДОСТУПНОЙ памяти
210 LET ПИП: I«PEEK 23730»25б*PEEK 23731 220 LET P = PEEK 23732* 256«PEEK 23733
гзо LET t - p - mm * i
400 REM »«««« Определение стартового адреса
4io PRINT -Lowest possible start - ":mm.., "Maximum space
avaiible = -": t 420 INPUT "Do YOU wish to chanet the lowest start address I
(V or H) ?":z* I 430 IF z*-"V OR z*^"r" THEN GO TO 7000 440 INPUT "Enter address at
      which to start loading machine
code":a
450 IF a<min OR 3>P THEN BEEP . 2,24: GO TO 440 500 GO SUB 8100 5io LET t-t a$min
520 PRINT "You can use цр to •; t" bYtes',,. 530 LET tt-PEEK 23675*256«PEEK 23676 540 IF
      a<u          AND U<P THEN PRINT 'if you use more than ";ц-а;"
bytes, you will overwrite the user defined graphics
```

```
area. " 550 IF a>^U THEN PRINT "You will overwrite the user defined
graphics area. "
560 INPUT "IS that OS (V OГ H) ? ";z* 570 IF Z*-"H" OR Z»="»l" THEN GOTO 7000 580 IF
      z*o"Y" AMD z$<>"Y" THEN BEEP .2.24: GOTO !>bO 700 REM »*« GO AHEAD          AND
      LOAD 710 LET l=a 7SO GOSUH 8SOO
760 INPUT "Enter number, b, f, i. d or t "; z* 770 IF z*:"" THEN BKEP .2,24: GOTO 7&O
      780 LET at-iCHR$ (CODE Z* (1 > 32" (Z*C II >"*- 1) 790 GOTO вOO'2OO*(a$-
      "B")*3O9«(a$-"F")*4CQ*(a$-"I"J'5GO«ld$
"D") i-6oo« (a$-*T")
SOO LET X-VAL Z*
S10 IF I>P THKH HEEP .2.24: GOTO 750
S2O IF JJ<0 OR K>256 OR ROIBT S THE» BEEP . a. ,'A : GOTO 760
S30 POKE 1.s
S4O LET 1 - 1 * 1
850 GOTO 740
1000 REII »«« перемещение вперед
IO1O I.KT 1-1-1
1020 IF LEH Z*>1 THEN LET 1-1M-VAL Zf(2 TO)
1030 IF l<a THEN LET I^a
1040 GOTO 7W
itOO BEH *«* Церемеиеиие назад
1110 LET 1=1*1
1120 IF LE» Z*>1 THEN LET 1-1 1*VAL Z*(2 TO)
1130 IF 1>P THEN LET I-P
1140 GOTO 740
1200 REM * * * вставка
1210 IF LEJI Z*-l THEN LET П-i: GOTa 1225
1220 LET n^VAL Z*(2 TO): IF П< 1 OR n>.p- I OR n<> INT n THEN
BEEP . 2.24: GOTO 740
1225 CLS:  GO SUB 8100: PRINT TAB b;"instrtme in progress" 1230 FOR j^p TO !>n STEP 1
      1240 POKE J. PEEK (J-П) rf 1250 NEXT J 1260 FOR J-l TO l*n-l 12TO POKE J.O 1280
      NEXT J 1290 GOTO 740 1300 REM *•* Удаление
1310 IF LKH Z*-1 THEN LET n I: GOTO 1JJO 13го LET n^VAL Z$(2 TO ): IF n<l OR n>P 1 OK no
      ГНТ n THKH
BEEP .2.24: GOTO 7ЧО \ 13 JO IF n<0 OR n>P-l THEN HEEP . 2, 21: GOTO ] Ji'O
1 34-0 CI.S :  GO SUB 6100 : PRINT TAfJ 6: "DELETfHG 1H MKXaeKb'i- I 1350 FOR J-I TO P n
      I 1J5O POKE J, PKKK (J*ni L.
```

MACHIHE CODE LOADER

ADDRESS	DECIMAL	CHECK SUH
32000	33	33
32001	0	33
32002	64	97
32003	1	98
32004	0	98
згоо5	24	12г
32006	гг	144
32007	J55	399
32ooe	122	sat
32009	150	671
32010	119	790
32011	35	625
32012	11	836
32013	120	956
32014	177	1133
32015	32	1165
3J016	247	1412
32017	201	1613

РИС- ВР1.

представлен экран во время работы загрузчика машинного кода-програнма "Screen Invert" загружена с адреса 32000.

5. Подпрограммы сдвига.

5.1 Сдвиг атрибутов влево.

Длина: 23

Количество переменных: 1 Контрольная сумма: 1574-назначение: эта программа сдвигает атрибуты всех символов экрана влево на одно знакоместо, переменные: Имя: new attr Длина:] Ячейка: 2329Б

комментарии: это атрибут, вводимый в крайнюю правую колонку. Вызов программы:

RANDOMIZE USR адрес контроль ошибок: Нет комментарии: Эта программа полоз на для выделения области тек ста или графики. Для прокручивания только 22 верхних строк 2Чг« должно быть изменено на 22.

```
ЛИСТИНГ НАЯННННЫХ КОДОВ НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,22528 33 0 08
            LD A, (23296J 58 0 91
            LD C.24 14 24»
NEXT^L            LD B. 31 6 31
NEXT_C            INC HL 35
            LD E, <HL] 94
            DEC HL 43
            LD (HL).E 115
ГКС HL 35
            DJNZ. NEXT_C 16 249
            LD <HL).A 119
            INC HL 35
            DEC C 13
JB HZ.NEXT_L 32 242
```

```
1370 NEXT J
1380 GOTO 7tQ.
140в STOP
1401 PRINT AT El.7;"PROGRAM TERHIHATED"
1410 STOP
7000 REM «*» Изменение RAMTOP
7010 INPUT "ENTER HEW START ADDRESS ":a
7020 IF 3<27000 OR a>P THEN BEEP .2,24: GOTO 710
7030 CLEAR a-1
7040 RUH
7999 STOP
8100 CLS
8110 PRINT TAB 6; "MACHINE CODE LOADER*,,.
8120         RETURN
8200 REM к» вывод содержимого памяти
8210  GO SUB 6100
8220 PRINT 'ADDRESS             DECIHAL CHECK SUH'
8230 LET C=0
6240 LET 3-1-6 : IF s<a THEИ LET s=a : GOTO 8280
6250 FOR J=a TO S-l
6260 LET C-C+PEEK J
8270 NEXT J
8260 LET f^S+17 : IF f>P THEN LET f-V
6290 FOR J-3 TO f
Д300 LET c-C+PEEK J
8310 PRINT AT j-s + 3.l;J: TAB 1Z:PEEK J; TAB 22; c
8320 NEXT J
8400 LET P03^1-3*3
8410 PRINT AT P03, 12: FLASH 1; PEEK ]
8420         RETURN
```

Как она работает:

в пэру регистров HL загружается адрес области атрибутов. аккумулятор загружается

значением ат рибута. вводимым в правую колон ку. в регистр с загружается количество строк для сдвига - он теперь может быть использован, как счетчик строк. в регистр в заносится число на 1 меньшее, чем число символов в строке, чтобы он использовался, как счетчик. HL увеличивается, чтобы указать на следующий атрибут. который загружается в е-регистр. HL уменьшается и во адресу нь помешается зна чение из е-регистра. HL увеличивается вновь, чтобы указать на следующий атрибут. Регистр в уменьшается и. если он не равен о. то происходит переход к NEXT.c'. HL теперь указывает на правую колонку и по адресу HL помешается значение из зккунулято-ра. HL увеличивается, чтобы указать на еле думу» строку- NEXT..L. Счетчик CTPQK <регистр с) унень-вается. Если значение результата не равно о. подпрограмма возвра-вается назад к NEXT.L.

по окончании работы программа возвращается в BASIC.

5.2 Сдвиг атрибутов вправо.

Длина: 23

количество переменных: 1 КОНТРОЛьная сумма: 1847 Назначение: Эта программа сдвнга-ает атрибуты все* символов экрана вправо на одно знакоместо.

Переменные: Имя: new attr длина: 1 ячейка: гзг*/б

комментарий: это атрибут, вводимый в крайнюю левую колонку. Вызов программы:

RANDOMIZE USR адрес контроль сшибок: нет Комментарий: Эта программа долез на для выделения области текста иди графики Для прокручивания только гг верхних строк 24» должно быть изменено на

2J

ЛИСТИНГ НАЖИННЫХ КОДОВ

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
1,D HL, 23295 33 255 86
            LD A.(23296) 56 0 91
            LD C.24 14 24¨
NEXT_L            LD B. 3! б 31
NEXT_C            DEC HL 43 S
            LD E, (HL) 94 j
            INC HL 35 |
            LD <HLJ.E 115 If
            DEC HL 43 II
            DJNZ. NEXT_C I6 249
            LD 1HLJ.A 119
            DEC HL 43
            DEC C 13
JK HZ,NEXT.L 32 242
            RET 201
```

Как она работает:

В пару регистров HL загружает ся адрес последнего байта области атрибутов. аккумулятор загружается значением атрибута для ввода в

яевую колонку. в регистр с загружается количество строк для сдвига - он используется, как счетчик строк, в регистр в заносится чис ло на 1 меньшее, чем число символов в строке для использования его в качестве счетчика.

HL увеличивается, чтобы указать на следующий атрибут, значение этого атрибута загружается в е-регистр. HL увеличивается, и по адресу HL помешается значение из е-регистра. HL уменьшается снова для указания адреса следушего атрибута. Счетчик в регистре в уменьшается и. если он не равен о, то - возврат назад к NEXT_c. HL теперь указывает на крайнюю левую колонку, и в ячейку с адресом Ни, помешается значение из аккумулятора. HL уменьшается для указания правого конца следующей строки. Счетчик строк уменьшается, и если он не равен о. - возврат назад к NEXT_L.

Программа возвращается в BASIC.

5.3 Сдвиг атрибутов вверх.

Длина: 21'

Количество переменных: 1 Контрольная сумма: 159» Назначение: Эта программа

сдвигает атрибуты всея символов экрана вверх на одно знакоместо, переменные: Инн: new attr Длина: 1 Зденки*: J8296

комментарий: это атрибут, вводимый в нижнюю строку. Вызов программы:

RAMDOMIZE USR ЗДРвС контроль ошибок: нет комментарий: эта программа полезна для выделения области текста или графики. Для дрокручи-вания только 22 верхних строк йй»« должно быть изменено на

160.

ЛИСТИНГ МАЛИННЫХ КОДОВ

```
HfcTKA АССЕМБЛЕР ЧЛСЛА ДЛЯ ВВОДА
            LD HL,22560 33 32 88
            LD DE, 22526 ЦТ 0 88
            LD BC, T36 1 224* г
LDIR 23T 176
            LD A,(23296) 56-0 91
            LD B, 32 б 33
NEXT,C 1..D (DEI, ft 18
ГНС DE 19
PJHZ NEXT_C 16 252
KET 201
```

Как она работает

В пару регистров HL загружается адрес второй строки атрибутов, в DE загружл< гея адрес пераой строки, и BC загружается числом байтов для перемещения,

количество байтов, определенное в регистровой паре вс, копи-

руются в ячейки памяти с адресом, находящимся в BE из ячеек, начинающихся с адреса в HL, используя инструкцию LDIS. Эти результаты в DE указывают на нижнюю строку атрибутов. аккумулятор загружается кодом атрибута для ввода в нижнюю строку. в-регистр загружается чи слон символов на одной строке- он используется, как счетчик.

По адресу DE помешается значение из аккумулятора, а затем DE увеличиваетея для указания на следующий байт. Счетчик уменьшается и, если он не равен о. подпрограмма возвращается к NEXT_c.

затем программа возвращается в BASIC.

5.4 Сдвиг атрибутов вниз

Длина: 21

Количество переменных: 1 контрольная сукна: 20ь7 Назначение: эта программа сдвигает атрибуты всех символов экрана вниз на одно знакоместо, переменные: Имя: new attr Длина: 1 ячейка: 2329& Комментарии: это атрибут, вво

диный в верхнюю строку. вызов программы:

RANDOMIZE USK адрес контроль ошибок: нет комментарий: эта программа полезна для выделения области текста или графики, для прокручивания только 22 верхних строк должны быть изменены: 223» на 159 255* на 191 224* ИЗ 160

-'«СТИНГ НАЙНННЫХ КОДОВ

```
METKA АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,23263 33 223 90
            LD DE,23295 17 255 90
            LD BC,736 1 224 2
LDDR 237 184
            LD A. (23296) 58 0 91
            LD B, 32 6 32
NEXT..C           LD (DE).A 18
            DEC DE 27
            DJNZ NEXT_C 16 252
            RET 201
```

как она работает:

В HL загружается адрес последнего атрибута ез-ft строки, в DE загружается адрес последнего атрибута 24-й строки. в BC загружается число байтов для перемещения, затем команда LDBR перемешает байты (их количество указано в регистровой паре вс) из ялреса в HL но адресу в DE, эти результаты в i)fc кранят адрес последнего атрибута первой строки.

в аккумулятор загружается значение атрибута для ввода в верх ню» строку, в в регистр загружа-

ется число байтов в верхней строке - он используется, как счетчик, в ячейку с адресом DE помешается значение из аккумулятора и 1)е уменьшается для указания на следующий байт. Счетчик уменьшается и. если он не равен о. подпрограмма возвращается к NEXT_c. Программа затем возвращается в

BASIC.

5.5 Сдвиг влево на один символ.

Длина:21

Количество переменные:О

Контрольная сунна:1745

назначение: эта программа прокручивает графику экрана на один символ влево, вызов подпрограммы:

KAHDOHIZE USR адрес контроль ошибок: Нет Комментарий:

эта программа полезна при организации экрана в качестве "ок на", показывающего в данный момент меньшую часть большой дисплейной области. Это "окно" перемешается, используя подпрограммы сдвига.

ЛИСТИНГ ИАШШШХ КОДОВ

```
НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,163S4 33 0 54
            LD D, L 85
            LD A. 192 62 192
NEXT_L            LD B, 31 б 31
NEXT-B            INC HL 35
            LD E. CHL) 94
            DEC HL 43
            LD (HL),E 115
            INC HL 35
            DJNZ NEXT_B 16 249
            LD (HL1.D 114
            INC HL 35
            DEC A F!
            JR HZ,NEXT_L 32 242
            RET 201
```

Как она работает:

В пару регистров HL загружается адрес дисплейного Файла, а D-регистр устанавливается во. в аккумулятор загружается число строк на экране. в в-регистр загружается значение на 1 меньшее, чем число символов в строке оно является числом байтов для копи рованкя.

HL увеличивается для указания на следующий адре<; и содержимое из этой ячейки загружается в е-регистр. HL уменьшается и в ячейку с адресом HL загружается значение из е-регистра. HL увели чивается для адресации следующего байта и счетчик в в-регистре уменьшается. Если он не равен о, подпрограмма возвращаете я к ыехт_в.

если регистр в равен о, это означает, что последний байт

строки скопирован и HL указывает на крайний правый байт, по этому адресу в регистровой паре HL по-невается о и HL увеличивается. указывая на следующую строку. Счетчик строк - аккумулятор уменьшается, и, если он не равен нулю. происходит переход к

NEXT.L.

программа возвращается в BASIC.

5.6 Сдвиг вправо на один символ.

длина:22

количество переменных:о Контрольная сунна:!976 Назначение: эта программа прокручивает содержимое дисплейного Файла на один символ вправо. Вызов подпрограммы:

RANDOMIZE USR адрес контроль ошибок: нет Комментарий:

эта программа полезна при организации экрана в качестве "окна", показывавшего в данный момент меньшую часть большой дисплейной области, это "окно" перемешается, используя подпрограммы сдвига.

ЛИСТИНГ НАЖИННЫХ КОДОВ

```
НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
1Л) HL, 22527 33 255 67
            LD D. O 22 O
            LD A, 192 62 192
NEXT.L            LD Б. 31 6 31
ЫEXT_B            DEC HL 43
            LD R,(HL) 94
            INC HL 35
            LD (HL).E 115
            DEC HL 43
            DJNZ NEXT_B 16 249
            LD (HL),D 114
            DEC HL 35
            DEC A 61
            JR HZ.KEXT.L 32 242
            RET 2(H
```

как она работает:

В пару регистров HL загружается адрес носледего байта дисплейного Файла, а D-регистр устанавливается в O. В аккумулятор загружается число строк на экране. В B-регистр загружается значение на 1 меньшее, чем число символов в строке. - он используется, как счетчик.

HL уменьшается, указывая на следующей байт, и его значение загружается в e-регнстр. HL затем увеличивается, и в ячейку с адресом HL загружается значение e-регистра. HL уменьшается, указывая

на следующий байт, и счетчик (в-регистр) уменьшается. Если он не равен o. подпрограмма возвращается к NEXT.e.

Если регистр в равен o, это означает, что HL указывает на крайний левыя байт строки, затем по адресу в регистровой паре HL яонеяается o, и HL уменьшается, указывая на следующую строку. Счетчик строк (аккумулятор) уменьшается и, если он не равен o. происходит переход к NEXT.L.

программа возвращается в BASIC

5.7 Сдвиг вверх на один символ

Длина: 68

количество переменных:o контрольная сумма: 6326 назначение: программа сдвигает содержимое дисплейного Файла вверх на восемь пикселей. Вызов подпрограммы:

RANDOMIZE USR адрес контроль ошибок: Нет Комментарий: нет

```
ЛИСТИНГ НАЖИМНЫХ КОДОВ
НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, 16364 33 O 64          LD DE,16416 17 32 64
SAVE        PUSH HL 229
            PUSH DE 213
            LD C. 23 14 23
NEXT.L            LD B, 32 Б 32
COPY_B            LD A, (DE) 26
            LD 1HL),A 119
            LD A,C 121
            AND 7 230 7
                                  CP 1 254 1
            JR HZ.NEXT_B 32 2
            SUB A 151
            LD (DE).A 18
NEXT_B            INC HL 35
            INC DE 19
            DJNZ COFY-B 16 241
            DEC C 13
```

```
            JR Z,REST 40 19
L» A. C 121
            AND 7 230 7
CF 0 254 0
            JR Z,H.BLOCK 40 22
                                    CP 7 254 7
            JR HZ, NEXT_L 32 225
            PUSH DE 213
            LD DE, 1792 17 0 7
            ADD HL,DE J5
            POP DE 209
            JR MEXT_L 24 217
REST        POP DE 209
            POP HL 225
            INC D 20
            INC H 36
            LD A,H 124
                                    CP 72 254 72
            JR HZ. SAVE 32 204
            RET 201
H-BLOCK           PUSH HL 229
            LD HL,1792 33 0 7
            ADD HL, DE 25
            EX DE,HL 835
            POP HL 225 I
J8 NEXT.L 24 198
```

Как она работает:

в пару регистров HL загружается адрес начала дисплейного Файла, а в ОБ загружается адрес байта через восемь линий вниз. HL и DE сохраняются в стеке, в С-регистр загружается число на 1 меньшее, чем число строк на экране, в В-регистр загружается количество байтов на одной линии дисплея -он используется, как счетчик.

В аккумулятор загружается байт, адресованный DE, и это значение загружается в ячейку по адресу HL, в аккумулятор загружается содержимое с-регистра и. если оно равно 1, 9 или 17. то в ячейку по адресу DE помешается о. HL и DE увеличиваются, указывая на следующий байт, счетчик в в-регистре уменьшается и. если он не равен о. происходит переход к

'COFY-B'.

Счетчик строк в регистре с затеи уменьшается. Если он равен о, происходит переход к 'RESTORE'. если с содержит 8 или 16, то происходит переход к 'H_BLOCK'. Если с не содержит 7 или 15. подпрограмма переходит к 'NEXT_L'. затеи 1792 прибавляется к HL - теперь HL указывает на следующий блок экрана, подрограина переходит к 'NEXT-L'.

в процедуре 'REST' DE и HL берутся из стека и 256 прибавляется к каждому из ннх. т. о., DE и HL указывают на строку, позиция которой ниже, чем та. что была в предыдущем цикле. если HL содержит значение 16432. подпрограмма возвращается в BASIC, иначе происходит переход к процедуре 'SAVE' . в процедуре 'H^BLOCK' 1792 прибавляется к DE - таким образом DE указывает на следующий блок экрана. Подпрограмма затем возвращается к 'NEXT_L*.

по окончании работы происходит возврат в бейсик.

продолжение следует

Алексеев А . Г ,.

МАСТЕРФАЙЛ 09 полная русификация.

ВСТУПЛЕНИЕ.

прежде всего я котел бы сказать, что эта статья ориентирова на на тек пользователей "спектру-ма", которые уже немного овладели програмированиен на Бейсике, защитом в ПЗУ компьютера но, erne не владеют знаниями, необходимыми для орогранирования в машинных кодах Z80. но, пользуясь только этини знаниями, многие усовершенствования можно уже делать с готовыми программами. а стимул для дальнейшего освоения иаяинныk кодов несомненно появится в процессе такой работы. Таким образом, психологический барьер будет преодолен и вы выйдете на новый уровень.

в этой статье изложены основ ные методы и приемы неполной и полной русификации программ для "Сдектрума". в качестве примера будут рассмотрены способы русификации программы "HF 09". она достаточно широко распространена среди пользователей "Спектрума" и английский вариант сильно сдерживает ее применение.

существует несколько способов русификации "Спектрума". сначала немного о саном простом из них.

1. Использование символов UDG-графики

Так как многие латинские буквы в режиме CATS LOCK по написанию совпадают с русскими (а. в. е. к, и, н, о, р. с. т, X), а вместо русской буквы "З" можно использовать цифру "3", то остается всего ао букв, требующихся для русификации (б. г. д. ж, и. и. а, п. V, ф. ц. ч, ш. щ. ъ. ы, ь. э, ю. я), то есть мы укладываемся в 21 сим вол, которые отведены для графики пользователя в "Саектруме".

Предлагаемая программа наглядно показывает принцип формировз ния символа UDG-граФики. например, для формирования буквы "я". закрепленной за символом UDG - "а", набираем программу:

ю LET n=USR "a" 20 FOR x^п то п»7 30 READ У tO POKE X. У 50 NEXT X

100 DATA BIN 00000000,

* BIN 00111J10.

BIN 0)000010.

BIN 01000010,

BIN oon mo,

BIH 00100010. B1Ы O1000D10, BIN O00Q0000

в строке 1оо после запятых надо набирать со 19 пробелов, чтобы наглядно просматривалась будущая буква, те пиксели, которые должны быть включены, отмечаем как "1". а те. которые выключены - как "о".

Для буквы "б" надо в строке 10 вместо USR "а" подставить USR "ь" и изменить нули и единицы в с троке 100.

Болев подробно этот метод был изложен в разработке инфоркома "Большие возможности вашего Спектрума". говорилось о нем и в

ZX-ревю к 4-5 (стр. 80). поэтому.

чтобы не повторяться, представим теперь, что русские UDQ-синволы вами уже сформированы и Вы записываете 21 символ, начиная с "а". на магнитофон:

SAVE "rilSUda" CODE USR "a".21»0

при наборе своих собственных программ можно поступать, например, следующим образом, после рестарта компьютера набрать:

```
1 C» TO 100
2 LOAD "ruSUdtf" CODE
3 GO ГО 1
5 SAVE "zagotowKa- LIHE г :
```

```
SAVE -rilSUde- CODE USR "a". 166
6 GO TO 5
```

получился своеобразный "де бют" программы, теперь сделайте RUH г и загрузите записанную ра нее область UDG. после сообщения -о ок" набирайте свою программу, начиная с той строки. которую указали в строке 1 после со то. то есть с 100, для записи ротовой программы на магнитофон исиоль эуите RUN 5. предварительно подставив вместо "zaeotowKa" имя программы. при этом запишется программа, а следом за ней блок кодов UDG. Строка б "зациклива ет* процесс записи, если Вам надо записать несколько дублей программы. при загрузке программа ав тостартует со строки S. и загрузит блок UDG. команда RUN обеспечим ет "холодный" старт программы.

блок кодов UDG графики, также. как и любоят другой блок кодов, можно разместить внутри Бей сик программы. используя для это го нулевую строку. это имеет определенные ареимушества: программа состоит не из двух, а из одно го куска, сокращается время загрузки. Делается это очень прос то.

После рестарта компьютера наберите: 1 REM. а после кем столь ко пробелов (или любых других символов), сколько байтов памяти вам надо зарезервировать для ваших целей, для размещения, например, блока символов UDG надо набрать 1бв пробелов. После того. как строка введена в память компьютера, заменяем ее номер на о. подав прямую команду: роке 23т5&.0. на экране теперь видим: 0 кен. Первая строка стала нулевой, теперь ее невозможно случайно испортить, вызвав на редактирование командой EDIT, свободная область в этой строке начинается с первого символа (пробела), стоящего за йен, адрес ее начала 23760. длина равна числу набранных символов (пробелов). Теперь, если вы набрали 1&6 пробелов, можете загрузить в нулевую строку блок кодов шк-графики:

LOAD "FUSUda" CODE 23760

Далее надо сделать так. чтобы включался блок UDG-кодов. зэгру женныв в новое место, для этого используем системную переменную UDG. занимающую два байта в ячей как 23675 и 23676. эта переменная указывает адрес первой ячейки области UDG. Выполните: PRINT реек й3675+25б"реек 23676

Вы получите результат: 65368. кстати, этот же результат вы получите . выполнив:

PRINT USR -a".

новое значение системной пере менной UDG будет равно гзтьо. вы числим младший, а затеи старший байты этого числа, выполнив:

PRINT 23760 256«1ят(23760/256> PRINT 1нт<2376о/25б>

получим 2Ой и 92. Теперь догрузим к нулевой строке наш "дебют" программы, выполнив:

MERGE "zaeotowKa" и изменим строки г и Ь: 1 GO TO юо

i' РСЖК 236Y5.200: FOKK Г3676. 9? 3 GO TO i

5 SAVE"zagotowKa" LINE 2

6 GO TO b

теперь сделайте RUN 2. после сообщения "о ок" можете набирать свою программу, используя графи ческий регистр. Надо только ном нить о том, что при выполнении команды LIST при выведении нуле? вой строки, скорее всего, будет сообщение об ошибке. это связано с тем, что интерпретатор Бейсика не понимает ту информацию, кото рая стоит в нулевой строке, для получении листин IM придется де лать LIST 1. Неприятности могут быть также и в том. что н<? все начальные строки могут быть видны в автоматическом листинге. хотя

они нормально зазываются при редактировании командой EDIT и работа готовое программы от этого никак не страдает.

для неслояння своих програю! этот способ русификации вполне может дрнненяться, так как позволит писать на русской языке даже коментарнн в тексте дрогранкы.

что касается программы HF от, то, к сожаление, к ней не подгодит этот самый простой способ русификации, так как символы, набранные с использованием графического регистра (курсор [61) не отображаются s режиме "дисплеи'. Вместо вхх на экране печатается знак "?". такие же трудности возникнут и бри выведении текста на печать при поиоши принтеров, отличных от "ZX". поэтому расСМОТРИМ другой, более совершенный способ неполной

русификации.

2. Использование дополнительного символьного набора.

некоторые проблемы, связанные с этим вопросом, были изложены в ZX-реис и 4-5 стр. 80. 81. Наибо лее крупная из всех - это. по-но-еиу. прогрэмная совместимость. Еакой латинской бгкве будет соответствовать какая русская? Это не имеет большого значения для конкретной игровой программы, где не надо вводить текст в процессе игры, однако для таких программ, как HF 09, это имеет немаловажное значение, ведь со временем все больше и больше появится возможностей пользоваться базами данных, составленными другими авторани, это могут быть и каталоги программ для "Спектрума" (с ко-нентариями на русском языке), и расписание движения поездов, самолетов и т. д. даже если Вы вместе с базой данных переплете и саму программу HF 09 автора. Трудности возникнут при внесении дополнений к изменений в базу данных, так как каждый раз надо будет "осваивать" незнакомое распределение русских букв между кнопками клавиатуры.

к стандарту кодов ASCII автор этих строк подошел своим, независимый от инфоркома путем, но результат все равно тот же. значит наверняка есть и другие авторы, работающие в этом стандарте. кстати и символьный набор "Спектрума" (коды с зг по \гт) являются кодами ASCII. Поэтому настоятельно рекомендую начинавшим придерживаться именно этого стандарта. Распределение русских букв на первый взгляд может показаться неудобным, по вы быстро привыкнете и перестанете это неудобство замечать, зато у вас будет больше шансов найти "привычные' для себя программы.

Базисная таблица кои-7 кодов

ASCII имеет несколько символьных наборов: латинский. русский, сне-шашй, символьный набор условно можно разделить на три части: 1 часть - коды с зг по 63 -

символы и цифры; г часть - коды с 64 по 95 -буквы, в основном печатаемые в регистре CAPS LOCK; 3 часть - коды с 96 по 187 -буквы, печатаемые без регистра CAPS LOCK. вы можете увидеть эти три части в трех строках на экране, выполнив:

FOR а=зг то 127: PRINT сня » а; : мехт а

То. что вы видите на экране, является символьным набором "но" базисной кодовой табдюш 5ои-7

кодов ASCII. если взять другой символьный набор: •Hi", то у него первая часть остается той же, вторая часть - это строчные русские буквы, а третья часть - заглавные русские буквы, таблица их соответствия с тем набором, который в вон "Спектрума" приведена в 2х-ревю к 4-5 за 1991г. . стр. 81.

когда вы принципиально решили для себя вопрос стандартизации, все остальное - дело техники, загружаем программу "ART STUDIO", входим в режим TEXT и далее роит EDITOR, теперь надо переделать латинские буквы на русские соответственно таблице (вторую и третью части символьного набора), оставив без изменения цифры и символы (первую часть символьного набора} и записать новый набор символов на ленту (например с именем "ruschr").

полученный символьный набор занимает 768 байтов в памяти, для использования в своих программах. его удобно расположить непосредственно перед символами UDG. разместив с адреса:

USR "а'-7бв=б4бОО

Кстати, сохранять на ленте деле сообразно символьный набор вместе с блоком кодов UDG; последние могут использоваться в Ваших программах непосредственно по назначение - для получения графических элементов, для этого надо об'единить эти блоки в один:

LOAD TUSChr- CODE б4600. 768 LOAD "ruaudfl' CODE 65368.168 SAVE TU3- CODE 64600.936

Теперь подумаем об удобстве пользования. Если символьный набор расположен с адреса 64600, то системная переменная CHABS (ячейки гзбОб, 23607) будет равна:

64600-г56-64344

нладшнй и старший байты ее:

64344-256»IBT(64344/2») =88 1HT(64344/256)=251

•дебют" программы может выглядеть теперь следующим образом:

1 GO TO 100

г LOAD •

3 GO TO 1

5 SAVE "zaaotowKa- LIKE г :

SAVE -rUS" CODE 64600,936

6 GO TO 5

8 POKE 23606,80: POCE 23607.

251: RETOBI : КвИ газ

9 P00 гЭбОС, О: POKE Z3607. 00: RETOR1 : ВЕЙ lat

Для записи готовой программы, как и раньше, используется вин 5 (вредваретедьно подставив имя программы), следом за программой будет записан символьный набор с

вЛОКОМ UDG (т. б. 768+168 = 936

байт), теперь поговорим о строках в и 9. подайте прямые команды: GO SUB а и затем: LIST Вы увидите, что листинг программы печатается русскими буквами, для обратного переключения сделайте GO SUB 9. теперь эти переключения ножно применять в вашей программе, например. чтобы написать фрззу: "zx-spectrum пишет по-русски*, надо ввести такт» строку в программе:

1оо PRI8T "ZX-Spectrun "; : GO SUB 8: PKINT 'Pllet PO-RUSSKI"

: GO SUB 9

чтобы не допускать ошибок при наборе большого об'ена русского текста в своей программе, перед набором строки сделайте GO SUB в пеной командой и вы будете видеть русский текст. который набиваете, в том виде, как он будет напечатан на экраяе.

применение двух символьны! наборов позволяет выводить на экран текст русскими и латинскими заглавными и строчными буквами, однако производить переключения, вставляя в программу GO SUB в и GO SUB 9 ае очень удобно, но главное, эти переключения нельзя выполнять, не останавливая программу, например, набирая строку данных в программе HF 09. поэтому ножно воспользоваться другим набором кодоа ASCII - это набор кон-7 'нс", здесь первая и вторая части - совпадают с кон "Спектру-на", а в третьей части вместо строчных латинских - заглавные русские буквы.

таблица соответствия с вон •Спектрума" приведена ниже (только третья часть символьного набора):

96 ФУНТ С 112 Р П

97 а А из q я

98 Ь Б 114 Г Р

99 С n 115 9 С

100 d Д 116 t Т

101 е Е 117 Ц У

Юг f Ф 118 Т I

103 в Г 119 V В

104 Ь Х 120 Х Ь

105 1 И 121 У И

106 J И 122 Z 3

107 К К 123 I Ж

юв 1 Л U4 верт. черта э

109 И И 125 I Ц

no n н 126 а востро* ч

111 0 0 127 КОПИРаЙТ ъ

в этом случае переключение на

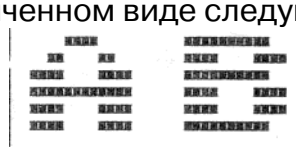
такой символьны* вабор можно сделать только один раз сразу же после старта программы, мзиетгге строку з нашего 'дебета":

з GO SUB в: GO то 1

при этой. правда. придется пользоваться хотя и русскими и латинскими, но только заглавными буквами.

попутно следует сказать несколько слов о том. как выглядит готовый нряфт на экране компьютера, вы обращали внимание, что почти ни в одной хорошея коммерческой программе не применяется символьный набор "Сдектрума" а. в основной, -стилизованные- ирмты. е особому зрительному з#4екту приводит применение "утолщенных" принтов, при этом программа бук-•ально преображается, почему бы не использовать такие шрифты в своих программах? кроне красивого "внешнего вида" они имеют повышенную четкость при различном сочетании цветов 1вк - PAPER,

в своих программах я применяю "утолщенный- русско-латинский символьный набор ("нс" в кодах еоя-7). который хочу предложить вашему вниманию. При этом буквы выглядят в увеличенном виде следующим образом (п - один пиксел) :



Однако большие массивы текста, состоящего только из заглавных букв, да еве "утолщенных", на экране выглядят слишком плотно и тяжело. Поэтому русские буквы в этом символьном наборе сделаны пониже, чем латинские, по высоте - как строчные буквы, а по написанию - как заглавные. при этом как бы увеличивается расстояние между строчками и текст выглядит более гармонично.

все то. о чен говорится в этих с'гроках. может бить подвергнуто сомнению. е тону же набирать 766 чисел вручную утомительно, но попробуйте, хотя бы из любопытства, я уверен: то. что Ы получите, вам понравится, одни явные преимущества этого символьного набора вы увидите сразу, а другие несомненно оцените, поработав некоторое время, пои друзья и зна-комж давно г с удовольствием пользуются именно этим набором, поскольку он достаточно универсален и текст замечательно выглядит на экране.

для ввода этого символьного набора в память компьютера, надо набрать программу

```
10 CLEAX 64599
го LET <=64600: LET s--o
x та$ J=i то 1«707
40 BEAD т: POCE ж. т
90 LET s=s»?:
60 ПОТ B
TO IF SO4465S THEM ШИТ FLASH
l:-Ottos-: STOP
во POKE гзбов.ав: рога гзб07,г51 90 го» л=зг то 127: PRINT сн»
A:: EEXT A
100 SAVI "ПИК" CODE 64500. 7в«
110 ПАТА
000. 000. 000, 000. 000, 000. 000. 000. 000. 024, 024, 024. 024. 000. 024, 000.
000. toe. toe. los, 000, 000, 000, 000.
000. 109, 294, 10в. 108. 254. 10в. 000
120 DATA
000. 024, 126. 088, 126. 026. 126. 024. 000. 098, 100. 0Св. 016. 0». 070. 000. 000. 04в.
      0вб. 048. 122.204. Ив. 000. 000. 024. 046, 000. 000. 000. 000, 000
130 DATA
000. 012. 024. 024. 024. 024. 012. 000. 000. 04в.024, 0г4. 024.024.048.000. 000. 000,
      10в, 096, 254. 0», 108. 000. 000. 000. 024, 024, 126. 024. 024, 000
140 DATA
000, 000. 000. 000. 000. 024. 024, 040. 000. 000, 000, 000, 124. 000, 000. 000. 000.
      000. 000. 000. 000. 024. 024. 000. 000. 000, 006. 01 г, 024.048. 096. 000
150 DATA
000. 060, 102, 110. 118. 102. 060. 000. 000. 024. 056. 024. 024. 024. 060. 000. 000.
      060, Юг. 006. 060.096. 126. 000. 000,060, 102.012.006. 102.060.000
160 DATA
000, 01г, 02в. 044, 076, 126, 012. 000. 000. 126. 096. 124, 006, 102. 060, 000. 000.
      060.096. 124, 102. 102. 060. 000. 000. 126. 102.012.024.048.048.000
170 DATA
000. 060. 102, 060, 102.102.060. 000. 000. 060. 102. 102. 062. 006. 060, 000. 000, 000.
      024. 024. 000. 024, 024. 000. 000, 000, 024. 024, 000.024.024.046
180 DATA
```

```
000. 000.012.024. 048.024.012. 000. 000. 000. 000. 124, 000. 124. 000. 000. 000. 000.
      04B. 024. 012. 024, 048. 000. 000. 060. 102.012.024.000.024,000
190 DATA
000. 124.206.214,222. 192. 124. 000. 000. 060. 102. 102, 126.102. 102. 000. 000.
      124.102, 124. 102. 102.124.000. 000. 060. 102. 096.096. 102. 060. 000
200 DATA
000. 124. 102. Юг. 102. 102, 124, 000. 000. 126.096. 124.096.096.126. 000. 000. 126.
      096. 124. 096. 096. 096. 000. 000.060. 102.096. НО. 102,060,000
210 DATA
000. 102. юг, 1гб. юг. юг. юг.000.
000. 060, 024, 024. 024. 024. 060. 000.
000. 014.006. оов. юг. юг.ово. 000.
000. 102. 108. 120. 120. 108. 102. 000
220 DATA
000.096, 096.096. 096.096.126. 000. 000. 066. 102. 126. 102. 102.102. 000, 000. 102.
      102, Ив. ПО. 102. 102. 000. 000, 000, 102,102, 102. 102. 060. 000
230 DATA 000. 124,102. 102. 124. 096.096,000.
000. 0б0. 102. 102. 110. ПО. 062. 000. 000. 124.102. 102.124. Юв. 102. 000. 000.
      060.096. 060. 006. 102.060,000
240 DATA 000. 126.024. 024.024.024.024. 000.
000. юг. юг. юг. юг. юг. ово. 000.
000. 102. 102. 102. 102.060. 024. 000. 000.198, 198.214.214.254.068,000
250 DATA
000. 102.060.024.024.0БО. 102.000. 000.102. 060. 024.024.024.024. 000. 000.
      124.012.024.048.096. 124,000, 000.030. 024. 024.024.024.030,000
260 DATA
000.000.096.048.024.012. 006. 000. 000.120.024.024.024.024. iaO.000. 000. 024. 060. 090.
      024. 024. 024. 000. 000. 000. 000. 000. 000, 000. 000. 255
270 DATA 000.000. 220. 246.246.246.ггo. 000.
000.000. 060.102.126. юг. юг. 000.
000. 000. 124.096. 124. Ю2. 124. 000. 000.000.108. 108. 108. 108. 1гб. 006
гво DAT;,
000.000.060. Юв. 108. 108.254. 196. 000.000.126,096.124.096,126.000.
      000.000.124.214,214. 124.016. 000. 000.000. 062. 04в. 048.048. 048. 000
290 DATA 000.000. 102,060.024.060. 102. 000.
000.000. юг. юг, но, не. to.1, 000. 000. ог4. юг. юг. по. не. юг. 000.
000.000. 102. 108. 120. 108. 10?.000
300 DATA 000. 000.030. 054,054. 054. Юг. 000.
000.000.066. юг. 1ге, юг, юг. 000. 000.000. юг. юг. 1гб. юг. юг.000. 000. 000, обо. юг.
      юг. Ю2. обо, 000
310 DATA
000. 000. 126. юг. юг. Ю2. юг. 000. ооо.ооо.062. юг. Обг. 054, юг. 000. 000.000. 1г4.
      юг. юг. 124.096.000. 000. 000. Обо. юг. 096. юг. о&о. 000 зго DATA
000.000. 126. 024.024.024.024.000. 000, 000. 102. 102. 062. 006. 060. 000, 000. 000.
      Z14. 214. 124. г!4. ?ЛЧ. 000. 000.000. 124. Юг. 124. 102. 124. 000
330 DATA
000. 000, 096.096. 1г4. 102. 124, 000. 000. 000. 198. 198. 246. 2гг. 246, 000.
000.000.обо. юг.01г. юг. ОБО.000.
000,000. 198.214.214.214. 254. 000 340 DATA
000. 000. 1го. 01г. обо, 01г. 1го. 000.
000. 000. 198. 214. 211. 211. г55. 00J,
000.000. юг. юг.обг.006,006.000,
060.066. 153. 161. 1М. !^3. 066. 060
```

Числа 8 строках DATA напечатаны друг под другом дли УДОбСТНЛ чтения. Вы же можете набирать их так. как Вам удобно.

теперь о "встраивании" затру жаеного символьного набора в программу HF O9. после того, как символьный набор будет записан на ленту, поступаем следусшим обра зои. Разместим его непосредствен но перед основным блоком кодов •W09CODE" CODE 57328.6208. с ад роса 57328-768-56560. значение переменной CHARS тогда будет рае но: 56560-256=56304.

младший и старший байты CHARS:

56304-256*INT(56304/256J ^Л40

IKT (56304/256)^Л19

Далее делаем: CLEAR 56559

LOAD "ZnaK" CODE 5656Q.766 LOAD -HF09CODE" CODE 5T328.&20в

К записываем ГОТОВУЮ программу на дентт:

SAVE -HF09 R/L- CODE 56560,89T6

Теперь изменяем програнму-загрузчик "HF LOADER", Заменив CLEAR 57327 на CLEAR 56559.

Изменения, производимые в основной Бейсик-программе. Добавля ем строки, переключающие основной и альтернативный символьные наборы:

```
в POKE VAL "23606". VAL "3*0": POKE VAL -23607". VAL "219": RETURN : REM rus
9 POKE VAL "23606". HOT PI : POKE VAL -83607-. VAL "60": RETURN : REM lat
```

изменяем другие строки, подставляя в нужные места GO SUB ' е или GO SUB 9 и изменив начальный адрес и ДЛИНУ основного блока ко лов:

```
1 СЮ SUB VAL "в": GO TO USR VAL "ЬвгвЬ"
4020 GO SUB VAL'9": SAVE C*(TO VAL "10') DATA F*(): GO SUB VAL "в": GO TO USR R
4030 GO SUB VAL "9": SAVE C*(TO VAL MO") LIHE VAL "4035": SAVE "HF09 R/L- CODE
      VAL"56560", VAL "8976": GO SUB VAL "в": GO TO USR R
4035 LOAD "MF09 R/L" CODE: GO TO PI/PI
```

Остальные строки оставляем без изменения.

Описанные выше способы русификации относились только к вводимым данным, однако более высокой степенью русификации является перевод на русский язык и замена текстовых сообщении в программе, которые печатаются по-английски.

.1. Перевод программы на русский язык.

Пусть Вас не пугает то, что программа написана в машинных кодах. На самом деле все ненамного страшнее, чей в Бейсике, причем для того. чтобы сделать полный перевод программы на русский язык, не обязательно даже умение пользоваться какими-либо специальными программами --мониторами типа HONS или другими, не обяэятельно также знание шестнадцатиричной системы счисления. Необходимо только желание, немного терпения и аккуратности, да хоть немного знать английский язык или иметь словарь потолше. неплохо также, если Вы поработали с программой какое-то время, чтобы Вам понятен был смысл того или иного текстового сообщения.

Рассмотрим подробно процесс перевода программы на русский язык на цринере программы ик 09. Приобретенный опыт Вы сможете ис-пользовать для перевода других программ.

в машинный кодак процедуры вывода текстовых сообпений на экран могут быть самыми различными, но в любом случае сама строка символов, выводимая на экран, находится внутри программы, надо только найти ее и изменить коды символов, находящихся там. занося другие значения хотя бы при помощи роке.

Дня работы можно воспользоваться любой программой-монитором, которая есть под рукой. а можно за несколько минут набрать нужную црограмму на Бейсике.

Ниже приводится описание такого специализированного монитора. который поможет нам находить текстовые сообщения в программе, а также несколько автоматизирует процесс замены текста на русский.

```
ПРОГРАММА-НОНИТОР 1 BORDER 7: PAPER T: INK 0: CLS : GO SUB 8: GO TO Ю0 г CLEAR 50000
3 LOAD "CODE
4 GO TO 1
Ь GO SUB 9: INPUT "FILEHAHE": N*
6 SAVE H* LIHE a: SAVE H* CODE 56560.8976
7 STOP
8 POKE 23606.240: POKE 23607. 219:           RETURN : REM FUS
9 POKE 23606,0: POKE 23607.60:        RETURN : REM lat
10 LET B= PEEK A: LET C*=CHR$ B 15 IF B<32 THEN LET C*="?" 20         RETURN
loo INPUT -ADDRESS: ";N
200 FOK A=H TO 65535
210 REM
220 GO SUB J0
230 INPUT (TAB 0;A: TAB 9;B; TAB 13;C»; TAB 23>;C»: IF C$<>" THEN POKE A, CODE C»
```

```
231 REM INPUT (TAB 0: A; TAB 8: B; TAB 13,-C»; TAB 23); LIHE C»: IF C*o"" THEN POKE A,
      VAL C*
240 GO SUB 10
250 PRINT TAB 0;A; TAB в; в; TAB 13; c*; 300 NEXT A
```

В строке 15 непосредственно перед знаком вопроса надо нажать

-IHV. VIDEO" (CAPS SHIFT*4). а сразу же после знака вопроса

-TR. VIDEO- (CAPS SHIFT*3),

REM в строке 231 набран после набора всей строки, в последаю» очередь.

после того, как вы наберете эту программу, сделайте RUN 2 и загрузите с магнитофона коды "HF09 R/L" с пристыкованнын русс ко-латинским символьным набором. После окончания загрузки программа выполнит переключение на этот символьный набор и запросит ад рее, с которого хотим просматривать содержимое памяти компьютера, пока введите "STOP-

(SYMBOL SHIFT+A) и "ENTER- и по говорим подробнее о самой программе-мониторе.

здесь вам уже знакомы некоторые фрагменты: строки в и у пе-

реключатели шрифтов (в программе проставлены в нужных местах GO SUB в и GO SUB 9>. аля сохранения результатов вашего труда вы периодически будете делать RUN 5 - это запись на ленту програнмы-ио-нитора и кодов HF 09 в том состоянии, до которого Вы дошли, после того, как сделаете RUH 5, программа запросит имя Файла для записи. Ножете задавать 1. 2, 3... и т. д.

строки to. 15. 20 - это вспомогательная подпрограмма, присва ивашая значения переменный в соответствии с содержимым ячейки памяти, строка 15 предохраняет от вывода на экран управлявших символов, имеющих коды со по 31. Если встретится такой символ, то на экране будет инверсно напечатан знак "?". Так вы сможете различать управляющий символ и настоящий знак вопроса, которые тоже будут встречаться в тексте.

строка 210 зарезервирована для организации поиска нужной информации, к ней мы erne обратимся позже.

строки 230 и 231 очень похожи. в строке 231 стоит REM и она, таким образом выключена, строка 230 выдает на экран адрес ячейки памяти, код. содержащийся в ней и символ, соответствующий этому иоду; далее - в кавычках ожидается ввод символа, код которого надо записать в эту ячейку, это удобно при замене текстовых сообщений, так как ввод осуществляется непосредственно буквой. на место того символа. который мы видим, измененный текст тут хе отображается на экране, удобно также и то. что переключившись на загружаемый символьный набор

Программы "HFO9 R/L". мы 6уд1-н

видеть текст в том виде, каким он будет в готовой программе, в пежине курсора 11,1 печатаем ру:;:-кие буквы, а в режиме tcj (CAPS LOCK) - латинские, действуют т.ж-же регистры "кхт. HODE- и -GRAPH". Вводить новый текст надо по одной букве, не забывая после каждого символа нажинать "ENTER", а также ставить "пробелы" между словами. Если вы ничего не хотите менять. то просто нажимайте "ENTER- для перехода к следующей ячейке, для того, чтобы остано вить программу, надо нажать "курсор влево" (SYMBOL SHIFT'S), за Тем "STOP" и "ENTER". потом опять

RUH или со то 100 для работы с новыми адресами. если же вы увидели. что допустили ОЩИбКУ при вводе, то остановите программу, и сделайте GO TO 20O программа стартует без запроса с того адре-га. который Вы вводили последлнй раз. нажимая "ENTER", подойдите к тону месту, где допущена ошибка. Теперь поэкспериментируйте с подготовленными программами, а в следующей выпуске ZX-PEBC ни доведен до конца рассказ о том. как выполнить полную русификацию

Програимы HASTERFILE 09.

СОВЕТЫ ЭКСПЕРТОВ

STALINGRAD

CCS 1989 г.

Эксперт Захаров М. Ю. г. Казань

Введение.

Стратегическая игра "STALINGRAD", разработанная фирмой CCS в 1989 году, представляет дальнейшее развитие игры "OVERLORD", расширяя ее возможности. Таким образом, фирма создала модели двух важнейших сражений второй мировой войны, переломивших ее ход на Западе и Востоке. Однако, если в "OVERLORD" Вы управляли войсками союзников, то в "STALINGRAD" Вам предлагается управление немецкими войсками. Можно не упоминать лишний раз о "холодной войне" и "образе врага" - одним словом, нам, конечно, это досадно и единственное, что можно сделать - подождать, пока отечественные программисты напишут программы, адаптированные под наше понимание истории.

Сражения на советско-германском фронте, уникальные своим огромным пространством и большим напряжением сил, драматичностью, наверняка еще послужат базой для сюжетов новых стратегических игр (например сражения за Москву, Берлин, сражение у озера Балатон, форсирование Днепра, освобождение Крыма, оборона Ленинграда, операция "Багратион", наступление в Маньчжурии), реализующих и советскую сторону в войне.

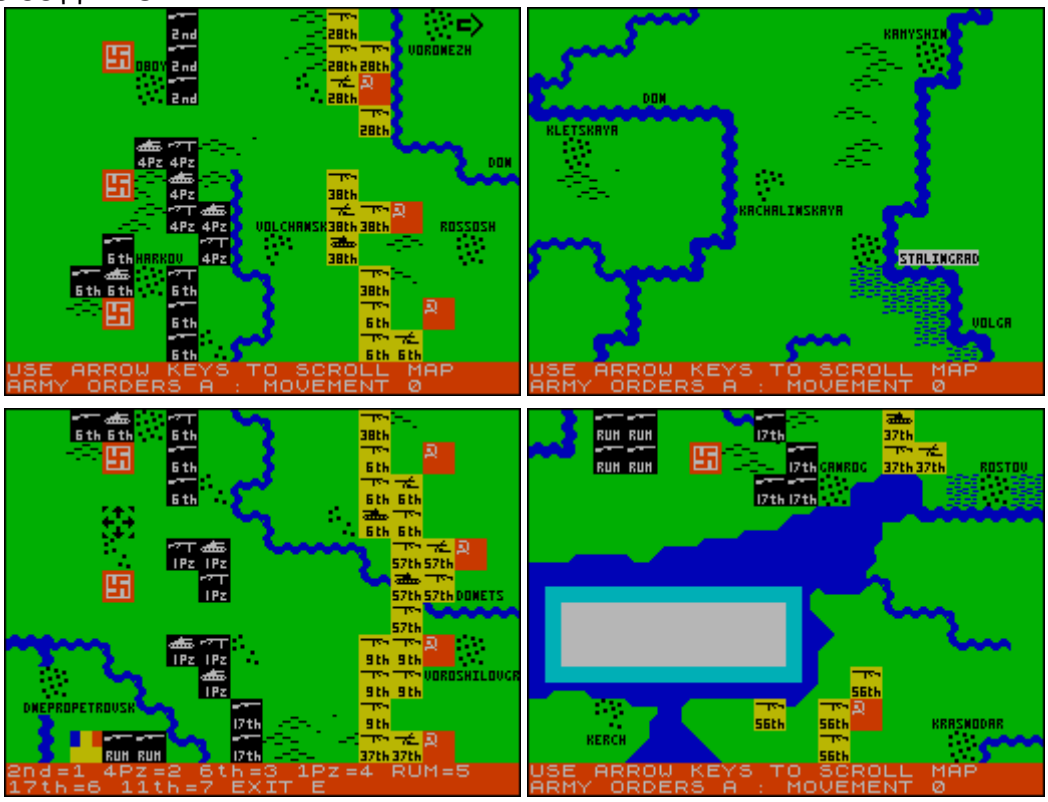
Вообще же, "STALINGRAD" - очень интересная игра, в ней реализованы такие возможности, как снабжение войск, корпусная организация армий, раздельное управление дивизиями (в OVERLORDe можно было управлять только армиями в целом). Боевые действия зависят от многих факторов (морального состояния войск, системы коммуникаций, рельефа местности и т. д.).

Она отражает события, происходившие на южном фланге советско-германского фронта с июня по декабрь 1942 года. В их результате Германия утратила, наконец, стратегическую инициативу, и перешла к обороне, как на Восточном, так и на Западном Фронте.

К началу лета 1942 г. Германия была еще настолько сильна, что перешла в наступление на половине своего Восточного фронта. Цели этого наступления были весьма решительны: планировалось пересечь Кавказский хребет, завладеть нефтяными промыслами Азербайджана и нависнуть над английским Ближним и Средним Востоком, Ираном и Индией. Как сопутствующая цель рассматривался захват Сталинграда и перехват Волги, как важнейшей транспортной артерии, связывающей Советский Союз с предстоящим районом боевых действий, а заодно с бакинской нефтью и южными маршрутами поставок союзников, Красная Армия еще не обладала к этому моменту силами, достаточными для отражения такого наступления, хотя и пыталась его упредить. Попытки кончились для нас катастрофами под Харьковом и Керчью.

Наступая, немцы заняли Донбасс, затем перевалы Главного Кавказского хребта, Моздок, вышли к Сталинграду, но нараставшее сопротивление советских войск заставило Гитлера отложить свои далеко идущие планы, сосредоточившись на Сталинграде, Сталинград постепенно приковал все его ударные силы, и там они большей частью нашли себе могилу. Накопившая силы Советская Армия сама перешла в наступление, и 23 ноября вокруг сталинградской группировки немцев замкнулось кольцо. Примерно этим периодом, судя по привлекаемым силам, завершается "STALINGRAD". Сталинградская битва же продолжалась. Советские войска отразили попытки немцев деблокировать окруженных, и к 2 февраля ликвидировали "котел".

Вы начинаете игру на позициях, примерно соответствующих линии фронта в июле 1942 г. В Вашем распоряжении 6 армий, одна из них в резерве - 3-я румынская (настроения в ней пониже, чем в немецких армиях). 11-я армия находится в Крыму. Вам противостоят слабо вооруженные и плохо оснащенные советские армии 1-го эшелона. Они пытаются перейти в наступление, но быстро терпят поражение. Этот этап вообще достаточно прост: Вы действуете вблизи баз снабжения, настроения в войсках высокие. Главная ударная сила в игре танковые дивизии. У Вас они сведены в 2 танковые армии, используйте их для быстрой победы. Попав хотя полуостров, вы активизируете свою 11-ю армию: здесь высаживаются 3 ее дивизии.



Однако кончается лето, начинается распутица, линии Ваших коммуникаций растягиваются. Вам приходится прикладывать все большие усилия для овладения новыми пунктами для баз снабжения. Численность ваших войск сократилась, а выделяемые верховным командованием пополнения скудны. Настроение в войсках падает. А тем временем из глубины Советского Союза подходят и разворачиваются свежие и хорошо оснащенные армии, имеющие большое количество танков и мотопехоты. (Обратите внимание на стрелки на краях карты военных действий: они обозначают полосы, в которых происходит это выдвижение), теперь все далеко не так просто!

Управление игрой

После загрузки программа запрашивает уровень сложности, предлагает загрузить отложенную партию, после этого Вы видите карту с расположением войск. Карту можно скроллировать клавишами курсора. В нижней части экрана находятся сменяющие друг друга меню.

Каждый ход состоит из 2-х этапов: отдания приказов и собственно перемещения войск. В начале каждого хода программа предлагает записать ситуацию (SAVE GAME - S),

отдать приказы (ARMY ORDERS - A) и перейти к их исполнению и перемещению войск (MOVEMENT - O).

Ваши армии состоят из 3-х родов войск: пехоты, мотопехоты, танков (каждый последующий род сильнее и маневреннее предыдущего). Аналогична структура советских войск, хотя, символы, обозначающие род войск различны.

Обратите внимание на символы с 4 стрелками. Это базы снабжения. Они могут располагаться лишь в населенных пунктах, являющихся узлами дорог. По мере продвижения Ваши поиски будут опираться на сеть этих баз. База появляется в очередном занятом Вами пункте, когда он оказывается у Вас в тылу.

Нажав в главном меню "A". Вы попадаете в меню принятия решений. Здесь Вы можете отдать приказы (ORDERS - O), просмотреть состояние сил обоих сторон (DETAIL - D) и их расположение (TERRAIN - T). Каждой армии соответствует цифровая клавиша, с помощью которой Вы к этой армии обращаетесь. Перейдя в режим ORDERS, Вы отдаете приказ выбранной армии, указывая ей рубеж, который она должна занять, и характер действий (наступление ATTACK - A, оборона DEFEND - D). Большинство Ваших армий, кроме 2-й и 11-й, включают по 6 дивизий, сведенных в 2 корпуса (3 дивизии в каждом), 2-я и 11-я армии состоят из одного корпуса каждая. Вы указываете последовательно положение центра, правого и левого флангов сначала одного, затем другого корпуса армии. Дивизии при этом помечаются символами "X", "R", "L" - то есть центр, правый и левый фланги корпуса. Таким образом, Вы можете указать желаемое место для каждой дивизии, что, по сравнению с OVERLORDом, большое удобство.

Прелесть же "самовольных" последующих действия дивизии, обусловленных принадлежностью ее к корпусу и армии, STALINGRAD унаследовал у OVERLORDa. Заданный рубеж может быть достаточно далеким: соединение будет выполнять свою задачу решая тактические вопросы с определенной долей самостоятельности. Однако если оно подверглось удару противника и стало откатываться назад, приказ (если Вы не отменяете его) лучше продублировать.

В режиме "DETAIL" Вы можете просмотреть численность войск обоих сторон. Вы увидите и настроение своих частей:

EXLT - excellent - превосходно
V. G. - very good - очень хорошее
GOOD - хорошее
FAIR - благоприятное
POOR - ухудшенное
LOW - низкое
ABYS - отчаяние

От настроения зависит эффективность ваших войск, их потери. В режиме "TERRAIN" программа выделит расположение выбранной вами армии. Ее дивизии закрашиваются одноцветными квадратиками без всяких надписей, что доставляет большое неудобство, поэтому этим режимом лучше не пользоваться. Выбравшись из младших меню в главное посредством многократных нажатии "E" (Exit). Вы и можете привести войска в движение, нажав "O" (MOVEMENT). Программа поочередно показывает движение и нанесение ударов обеими сторонами (удары наносят, разумеется, только наступавшие части). Вы видите потери, которые несут сражающиеся дивизии в процентах. Между действиями немецких и советских войск Вам предлагается пополнить свои части. В рамке-меню, где-то в районе Азовского моря, перемещая указатель клавишами "P" и "Q", Вы выбираете очередную пополняемую армию, нажимая "A". Если армия отрезана от баз снабжения, доступа к ней нет. Затем - род войск пополнения (Фиксация - "T") и его численность (SET VALUE - V). В этом меню Вы видите размеры имеющихся у Вас резервов:

ARM - танки
MECH - мотопехота
INF - пехота (инфантерия)

На протяжении нескольких первых ходов пополнения Вам не выделяются.

В ходе боевых действии некоторые дивизии не просто несут потери, но исчезают совсем: они могут быть расформированы (Unit disband) или разгромлены (Unit routs). Остатки расформированной дивизии пополняют другие дивизии армии. Вместо исчезнувших дивизий можно сформировать новые из пополнения любого рода войск. Таким образом, в состав пехотных армий можно включить танковые и мотопехотные дивизии, и наоборот (но не стоит распылять силы).

Формирование новой дивизии произойдет, если передать армии большое пополнение - 100 или 200.

Армия не может превзойти своих первоначальных размеров: она не может включать больше 6 дивизий, а дивизия не может иметь более, чем 100% состав.

Когда все Ваши резервы будут исчерпаны, программа выдаст Вам свое заключение о том, кто же победил, об устойчивости победы и сообщит количество выживших в результате Ваших действий у обеих сторон. Заключение вполне может не оправдаться, Вы можете продолжить игру, ответив "Y" на соответствующий вопрос. Правда, игра с каждым ходом будет все больше напоминать эндшпиль шахматной партии, корда на доске остается по 2-3 фигуры.

Некоторые детали

Пусть Вас не удивляет, что последние советские армии первого эшелона стремительно расформировываются, выйдя из соприкосновения с Вами. Они очищают место для вступления на сцену II эшелона, вливаясь в его дивизии.

Обратите внимание на то, что программа перемешает армии последовательно, по их расположению с севера на юг вдоль линии фронта. Поскольку порядки одной армии плохо проницаемы для другой, это правило стоит учитывать, если Вы не хотите, чтобы Ваши войска перепутались.

Компьютер скрывает расположение своих частей, не находящихся в контакте с Вашими. Но, если Вы вызовете данные о них в режиме "DETAIL" или "TERRAIN", он проскроллирует карту и укажет приблизительный район их расположения, хотя, конечно, цветом их не выделит.

Следите за целостностью своих коммуникаций. Снабжение армии может быть прервано по нескольким причинам: она далеко оторвалась от баз снабжения, советские войска совершили рейд по Вашим тылам (нередко компьютер направляет Вам в тыл танковые бригады), наконец, Вы сдали свою базу, отходя. Лишившись снабжения, армия сразу теряет в эффективности: падают маневренность, настроение в войсках, растут потери и их нечем восполнять. Для разгрома такой армии достаточно нескольких ходов. Чтобы восстановить снабжение, надо заново создать всю систему коммуникаций, для чего армия должна вплотную подойти к любой базе снабжения. Нередко приходятся довольно долго отступать.

Уровни сложности игры

В начале игры программа предлагает выбрать уровень сложности (ENTER GAME LEVEL). Их три: (1-3).

Принципиальных различий между ними нет, но из-за многочисленных мелких сложностей игра на 3 уровне значительно труднее.

Выгрузка файлов на ленту

Игра достаточно продолжительна, полный ее цикл длится около 8 часов, поэтому Вам наверняка придется записывать партию на ленту, и не один раз. Программа предлагает сделать это в начале каждого хода. Перед выдачей информации на магнитофон выдается предупреждение (Press any Key). Записываемый файл имеет длину более 16 килобайт, в нем сохраняются не только положение войск, но и отданные Вами приказы.

Должен предупредить об одной детали. Бывает, что программа выводит сообщение "Unit disband" уже после записи партии. В этом случае на моем компьютере версии "Балтика" записанный файл оказывается сбойным. После его загрузки игра "виснет".

Приходится дожидаться следующего хода и записывать партию заново.

Программа предлагает загрузить отложенную партию только один раз, в начале игры, после ввода уровня сложности.

STALINGRAD принадлежит к тем немногим играм, которые не могут быть запущены заново без перезагрузки из-за большого объема информации, характеризующей ситуацию. Поэтому нельзя прервать неудачную партию иначе, как только перезагрузив программу. По достижении победы одной из сторон программа тоже предлагает считать себя заново с ленты.

Замечания

События, реализуемые игрой, начальное положение сторон, состав сил, стартовые действия, завязывающие ситуацию, конечно, лишь напоминают реальные события 50-летней давности. Все мои попытки повторить Сталинградскую битву не привели к успеху. Советские войска в игре не включают ни 5-й танковой армии, ни танковых и механизированных корпусов, танки и мотопехота рассеяны по общевойсковой армиям, 2-я немецкая армия в действительности была венгерской. 11-я армия вполне самостоятельно переправилась из Крыма летом 1942 г. Существует и много других немаловажных деталей, отличающих Сталинград от STALINGRADa.

Но, как самостоятельная игра, STALINGRAD очень интересна и наверняка привлечет ваше внимание.

FLIGHT SIMULATION[1]

Psion 1983 г.

Перевод фирменной инструкции "ИНФОРКОМ", 1987

В последние годы пилотов все чаще обучают, используя имитаторы. Даже на маленьком компьютере можно реально, в реальном времени, отобразить многие особенности полета: динамику самолета, навигацию, инструментальное обеспечение и т. д. "Имитатор полета" включает в себя эти эффекты и представляет модель полета маленького двухмоторного винтового самолета.

Аспекты полета

Органы управления самолетом включают в себя: штурвал, закрылки, хвостовой руль и регулятор мощности двигателей. Движение штурвала вперед и назад вызывает отклонение горизонтальных рулей и, соответственно, перемещение самолета вниз или вверх.

Аэродинамика самолета чрезвычайно сложна. Управление каким-либо органом вызывает, как правило, не одно последствие. Например, элероны не только вызывают крен самолета, но и создают дополнительные боковой воздушный поток, вызывавший поворот самолета. Всему этому можно научиться, работая с имитатором.

Положение самолета и характер полета отображаются на приборной панели в кабине

[1] Последующие версии носят название Flight Simulator (Прим. NUK)

пилота. Пилот должен руководствоваться показаниями этих приборов при выведении самолета на линию, с которой он совершит посадку с требуемой скоростью, под необходимым углом. Обычно правильный угол посадки составляет 3 град., т.е. высота должна быть 6000 футов на расстоянии 20 миль, 3000 футов на расстоянии 10 миль и 1000 футов на расстоянии 3 мили от ВПП. Руль направления также может несколько изменять курс самолета. При движении по ВПП именно им регулируется направление движения самолета.

В имитаторе Вы можете приземляться на одном из двух аэродромов, взлетать с них, ориентироваться с помощью радиомаяков или по местности. Через кабину Вы видите светлое небо и темную землю, огни взлетно-посадочной полосы в трехмерной перспективе и объекты на земле: озеро и т.п. Экран можно переключить также на изображение навигационной карты, на которой показаны детали местности: ВПП, радиомаяки и пр. После загрузки программы в машину на экране появляется меню с вопросом: "Что Вы хотите отрабатывать?" - взлет, гладкий полет или посадку. Нажмите, соответственно, 1, 2 или 3. После этого Вас спросят, нужен ли вам учет ветра. Отвечайте "Y" только если Вы опытный пилот и можете справиться с внезапными порывами ветра, в противном случае - "N".

Приборная панель

На нижней части экрана изображена приборная панель кабины пилота. Пять циферблатов слева-направо это:

1. Система инструментальной посадки (ILS).
2. Датчик скорости.
3. Оборудование наведения на радиомаяки (RDF).
4. Высотомер.
5. Индикатор скорости высоты (ROC).

RDF - это большой циферблат в центре панели. Здесь изображен символ самолета, он указывает направление полета самолета. Цифровой указатель дает направление полета в градусах компаса. RDF -это самая необходимая навигационная система, в любое время полета самолет связан с одним из радиомаяков, положение маяка, к которому в данный момент подключился самолет, изображается светлым крестиком на окружности циферблата RDF. Если Вы хотите двигаться на данный маяк, Вам надо развернуть самолет так, чтобы положение маяка совпадало с "12 часами" на циферблате RDF.

Датчик скорости находится справа от RDF. Малая стрелка указывает высоту в тысячах футов, а большая в сотнях футов.

Индикатор ROC - это индикатор скорости подъема. Он измеряет вертикальную скорость самолета в единицах 1000 футов/мин. Когда стрелка выше нуля, самолет идет вверх и наоборот.

Индикатор POWER - "мощность" - измеряет тягу моторов, тяга двигателей возрастает при увеличении POWER, но на большой высоте падает.

FUEL - "топливо" - наличие топлива в баке.

FLAPS - "закрылки" - показывает угол выдвижения закрылков.

GEAR - "шасси" - имеет два состояния. Шасси убрано - красный цвет; шасси выпущено - зеленый цвет.

BCN, RGE, BRG - информация по маяку, к которому привязан в данный момент самолет. BCN - позывной маяка. RGE - расстояние до маяка в милях. BRG - курс в градусах относительно маяка. ILS -система инструментальной посадки. Эта система для наведения самолета на аэродром. Радиобуй, размешенный в начале ВПП, излучает сигнал, положение которого видно на экране ILS, как светящееся пятно. Когда самолет приближается к аэродрому с правильного курса, светящееся пятно будет в центре экрана.

Ra - радиовысотомер, это часть системы ILS. Радиосигнал, отраженный от земли, позволяет замерить высоту от земли до колес самолета. Замер в футах. Это точный отсчет: применяется при посадке.

Управление самолетом

Управление влево-вправо-вверх-вниз с помощью курсора - клавиши 5, 6, 7, 8.

Управление горизонтальным рулем - Z - поворот влево. X -вправо. Им же управляется самолет при движении со ВПП.

Тяга двигателя: P - повышение. O - понижение.

Закрылки:

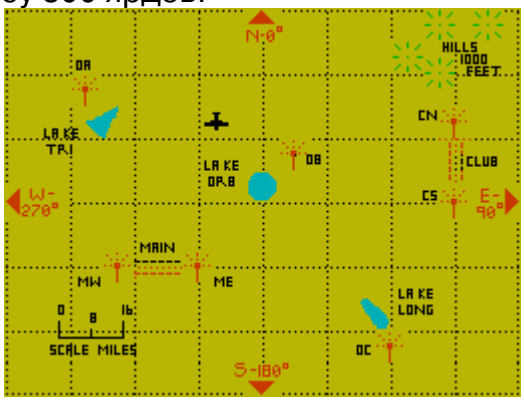
F - увеличение выхода закрылков; D - втягивание закрылков.

С убранными закрылками скорость отрыва самолета - 80 узлов, с выпущенными закрылками - 60 узлов. Увеличение выхода закрылков при большой скорости полета может привести к разрушению крыльев.

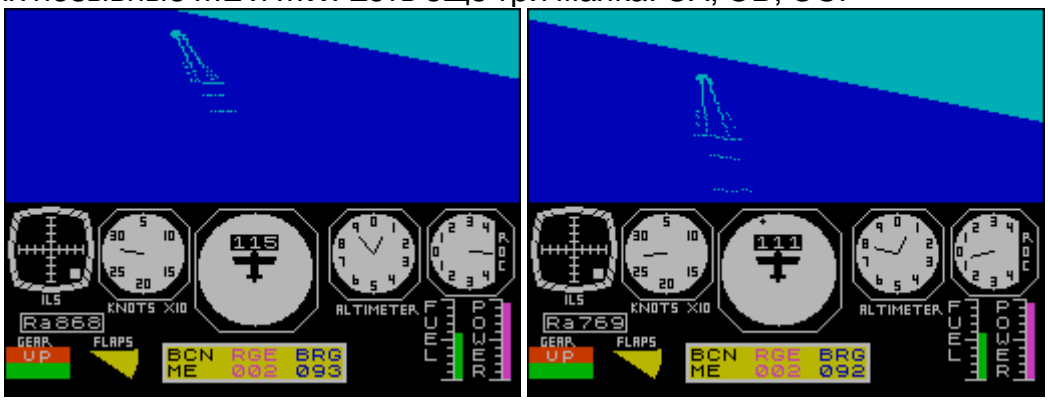
Шасси: выпуск и втягивание - клавишей G. Нельзя выпускать шасси при большой скорости, т. к. оно может заклинить или оторваться.

"Радиомаяки": - чтобы подключиться к другому маяку, нажмите клавишу B. Пока Вы будете держать эту клавишу, изменяется маяк, к которому привязан самолет.

Карта: - нажав кнопку M, Вы получите на экране изображение навигационной карты. На карте изображены два аэродрома: один международный аэропорт. MAIN и один маленький клубный аэродром CLUB. MAIN имеет длинную ВПП длиной более мили и поэтому посадка на него для небольшого самолета несложна, CLUB - это маленький местный аэродром и имеет полосу 800 ярдов.



Карта изображает также положение маяков и различных наземных объектов. Около главного аэропорта есть два маяка, расположенные на расстоянии трех миль от начала и от конца полосы. Их позывные ME и MW. Есть еще три маяка: OA, OB, OC.

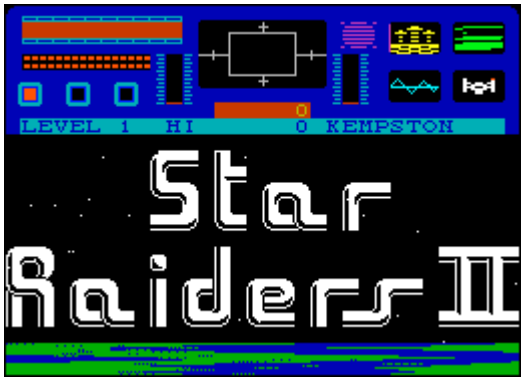


Навигация

Самая трудная часть полета это подход к аэродрому и посадка. На достаточно большой высоте Вы можете экспериментировать с органами управления, меняя скорость и направление полета, не заботясь о навигации. Если же Вы хотите совершить посадку, Вам надо положить самолет на правильный курс и подойти к аэродрому с правильным углом посадки. Это трудная задача и она требует больших усилий и тренировки, пока Вы совершите посадку правильно.

STAR RAIDERS[1]

"Звездные рейнджеры"
ATARI 1987.



Эксперт Порядин С. В. Марийская ССР

Игра STAR RAIDERS 2, разработанная фирмой ATARI, органично объединяет в себе несколько игровых жанров. В этой игре Вы можете проявить себя, как стратег в планировании операции против захватчиков из другой звездной системы, так и как пилот боевого космического корабля в схватках с эскадрами кораблей захватчиков.

Боевые действия ведутся как в открытом космосе в двух звездных системах: Вашей "CELOS" и вражеской "PROCYON", так и над поверхностями планет, на которые высадились вражеские корабли. В вашей звездной системе вокруг центрального светила вращаются три планеты, одна из которых имеет спутник. Все планеты и спутник заселены и основное население сконцентрировано в городах. Уничтожение городов является главной целью захватчиков.

В Вашей звездной системе также находятся несколько космических станций, на которых отремонтируют Ваш корабль, дозаправят его горючим и боеприпасами.

Враги

Они прилетают из своей звездной системы. Их корабли никогда не появляются в одиночку, они все время летают эскадрами, в состав эскадры входит несколько флайеров (иногда их число больше дюжины), два или три крейсера и один флагманский корабль.

Флайер имеет легкое вооружение и может быть сбит единственным попаданием, но флайеры нападают сразу вдвоем или втроем, так что неопытному пилоту будет поначалу очень "жарко". При уничтожении всех флайеров в эскадре противника наступает черед мощных и опасных крейсеров. Они имеют очень мощное оружие и могут быть поражены одним или несколькими попаданиями бомб. Смена оружия в вашем корабле происходит автоматически, как только на Вас нападет крейсер или флагманский корабль. Но космические битвы - это не то, для чего предназначены крейсера (хотя они очень маневренны). Основная их цель уничтожение городов на Ваших планетах.

Флагманский корабль гораздо мощнее вооружен чем крейсер, но опасности для Ваших планет он не несет. Правда, за уничтожение флагмана Вам будет начислена большая сумма очков. Но основная Ваша цель - это защитить города на планетах.

---

[1]Игра Star Raiders в природе не обнаружена. Все иллюстрации относятся к игре Star Raiders II (прим.OCR).

Вражеские эскадры также могут атаковать станции. Если врагов вовремя не отогнать от них, то станции могут быть уничтожены и Вы останетесь без баз. На нижнем уровне игры, когда у Вас имеется три станции, потеря одной из ник не не-сет больших неудобств. Но на более высоком уровне это может обернуться для Вас катастрофой.

Уничтожая вражеские эскадры, невозможно добиться победы, так как из вражеской звездной системы в Вашу будут лететь все новые и новые экспедиции. Чтобы одержать полную победу над врагом, нужно уничтожить его базы. Базы врага находятся на трех планетах его звездной системы. Уничтожить базу противника можно метким попаданием атомной бомбы. Для включения системы бомбометания при полете над поверхностью планеты необходимо нажать клавишу "W". Тогда внизу обзорного экрана появится прицел для бомбометания. При приближении к базе на медленной скорости нужно вывести ее изображение под прицел и нажать "огонь". За один заход невозможно уничтожить все базы противника, т.к. у Вас ограничен запас бомб, так что придется возвращаться на станцию для пополнения боезапаса.

Запуск игры

После загрузки программа предложит вам выбрать тип джойстика и игровой уровень от "1" до "3". После нажатия "S. SHIFT" Вы приступаете непосредственно к игре.

После запуска игры Ваш корабль находится на одной из планет своей звездной системы. Он сразу же будет атакован вражескими флайерами, но это лишь разведчики врага, и никакой опасности для городов они не представляют. Поэтому вы можете сразу отлететь от планеты, нажав "SPACE", выбрав курс и затеи нажав клавишу "огонь", теперь, в более спокойной обстановке, можете внимательно рассмотреть находящееся на экране Вашего дисплея изображение.

Экран

Весь экран можно условно разделить на две части, в верхней, меньшей его части, расположена приборная доска вашего корабля, на ней находятся несколько индикаторов, показывающих состояние его систем.

В левой верхней части расположен индикатор энергии, чуть ниже этого индикатора расположено два ряда прямоугольных меток - это счетчик атомных бомб, имеющихся в Вашем распоряжении. Ниже этого счетчика находятся три индикатора, показывающих, какое оружие в данное время Вами задействовано.

В центре приборной доски находится экран радара, который позволит ориентироваться в сложной обстановке боя. Если нажать клавишу "T", то изображение на нем сменится изображением Вашего корабля, на котором условными метками будут указаны системы, пострадавшие от вражеского огня. Если контур корабля на экране радара обнесен точками, это значит, что включено защитное поле, в местах, где поле пробито огнем вражеских кораблей точки отсутствуют или мигают.

Но обе стороны экрана радара находятся две вертикальные шкалы показывающие температуру лазеров, ниже расположен счетчик очков, заработанных Вами.

Остальная информация на приборной доске не несет большой смысловой нагрузки.

Ниже приборной доски расположен экран Вашего корабля, на котором и показывается обстановка по курсу. Между экраном и приборной панелью находится информационная строка, на ней в ходе игры будут появляться сообщения о различных

действиях врага и о критическом уровне энергии в Вашем корабле.

Если во время игры нажать клавишу "SPACE", то на экран корабля будет спроецирована карта звездной системы, в которой он находится. При этом внизу экрана расположена информация о планете, на которую Вы направили указатель гиперперехода или на который Вы находитесь.

Управление

Управление кораблем не очень сложное и осуществляется с помощью выбранного джойстика. При полете над поверхностью планеты движением ручки джойстика вверх-вниз регулируется скорость Вашего корабля.

Клавиша "S" служит для вкл/выкл защитного поля.

Клавиша "W" служит для переключения стрельбы лазером на бомбометание и обратно. На то, какое оружие используется, указывает соответствующий индикатор.

Клавиша "T" служит для переключения экрана радара на индицирование состояния систем корабля. При повторном нажатии включает радар.

Клавиша "P" служит для временного останова игры.

Клавиша "C. SHIFT" заканчивает игру и выводит на экран заставку.

Для гиперперехода необходимо вывести на экран карту звездной системы, в которой Вы находитесь, и, манипулируя джойстиком, нужно установить указатель гиперперехода на любой объект в этой системе. Если после этого нажать "огонь", не выходя из "карты", то переход будет сделан туда, куда Вы указали.

Перелет в другую звездную систему можно осуществить, предварительно установив указатель гиперперехода на ее символьное изображение в углу карты и нажав клавишу "огонь".

Энергия

Энергия расходуется на поддержание защитного поля во время обстрела Вашего корабля вражеским и во время гиперпереходов. При полном расходе запаса энергии корабль погибает.

Теперь несколько советов, которые могут быть Вам полезны.

Старайтесь не допустить высадки вражеской эскадры на планету в Вашей системе, так как сразу после этого крейсера начнут уничтожать города, да и сбить крейсер над планетой значительно сложнее.

Не старайтесь полностью уничтожать эскадру в космосе. Достаточно уничтожить все крейсера. Флагманский корабль, оказавшись в одиночестве, не представляет серьезной угрозы, а схватка с таким мощным кораблем после боя с крейсерами, когда в Вашем корабле мало энергии, может оказаться роковой.

При нападении на станцию вражеской эскадры, ее изображение на карте начинает мигать и Вам нужно спешить к ней на выручку, но если у Вас есть более срочные дела, например уничтожение высадившихся на планету врагов, то разборку с врагами, напавшими на станцию, можно отложить. Некоторое время станция может продержаться сама. Однако, следует помнить, что на более высоких уровнях игры это время значительно сокращается, да и станций там меньше (на третьем уровне - всего одна станция). Так что рисковать не советуем.

Из этой войны Вы выйдете победителем, если уничтожите все базы противника и все эскадры вражеских кораблей, но при этом у Вас должен остаться хотя бы один город на любой из планет. Если же враги уничтожат все Ваши города, то это будет поражение. Естественно, Вам будет засчитано поражение, если враги смогут сбить Ваш корабль. На первых порах это и будет происходить чаще всего, но, потренировавшись и накопив боевой опыт, Вы станете серьезным препятствием на пути космических агрессоров.

THE TRAIN

("Поезд")
Accolade 1988.
Эксперт Порядин С. В. Марийская ССР

The Train представляет собой очень оригинальную программу-иммитатор. В этой игре Вам будет предложено испытать свои силы в качестве машиниста паровоза. Кроме того, в ней присутствуют элементы других игровых жанров, в том числе и стратегических игр.

Краткий сюжет

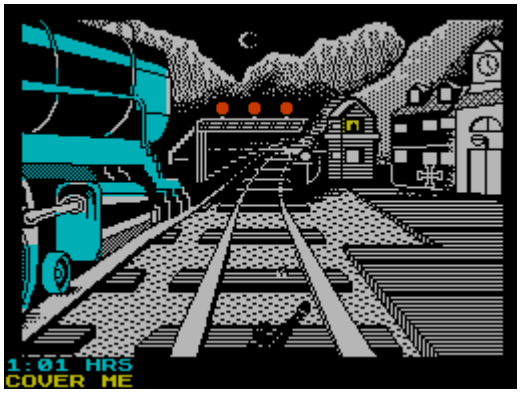
Представьте себе, что Вы находитесь в центральной части Франции в 1944 году. Войска союзников высадились на западном побережье и ведут бои с войсками вермахта. Вы командир одного из отрядов сопротивления, и Вашему отряду дано задание отбить на станции METZ (г. Мец) бронепоезд и провести его к войскам союзников.

От Вашего умения ориентироваться в сложной обстановке зависит успех данного предприятия. Вам нужно спланировать маршрут движения бронепоезда по железным дорогам Франции. По пути придется прорываться через станции и железнодорожные мосты, захваченные фашистами. Их нужно отбить у врага. В пути на бронепоезд будут совершать налеты самолеты Люфтваффе и Вам придется от них отбиваться. Ну и кроме всего этого, от Вас потребуется непростое умение справиться с такой сложной машиной как паровоз. Ведь кроме опасностей, подстерегающих Вас на пути, могут произойти большие неприятности связанные с неумением справиться с органами управления в кабине паровоза. Вы можете погибнуть от взрыва котла или, наоборот, бронепоезд остановится из-за нехватки давления пара, если вы неправильно рассчитаете маршрут движения и у Вас кончится уголь или вода. У паровоза также могут выйти из строя тормоза, и игра на этом закончится.

Настройка программы

После запуска программа предложит выбрать джойстик и сразу после этого Вы приступите непосредственно к игре.

В начале игры Вы находитесь с пулеметом на путях возле паровоза, ваша задача - прикрыть своего помощника Ле Дюка, который направился к стрелке, чтобы перевести ее. Вам нужно уничтожать немецких солдат, силуэты которых будут появляться в окнах зданий расположенных поблизости. Запас патронов у Вас неограниченный и можно порекомендовать не отпускать гашетку и стрелять длинными очередями.



Как только Ле Дюк доберется до семафора, Вам будет предложено выбрать путь, по которому пойдет бронепоезд. Путь выбирается с помощью клавиш управления или джойстика. Затем нужно прикрыть возвращение вашего помощника. Как только он доберется до паровоза, Вы автоматически оказываетесь на рабочем месте машиниста и приступаете к основной части игры.

Экран

Перед Вами расположено несколько приборов и органов управления паровозом. На приборах показывается: скорость паровоза (MPH), давление пара (PSI), его температура (TEMP) и уровень воды (WATER).

К органам управления относятся: тормоза (BREAKE), рычаг реверса (FORWARD/REVERSE LEVER), заслонка сброса пара (STEAM BLOWOFF), клапан регулировки давления (THROTTLE), кроме этого перед Вами в кабине находятся крышка топки и ручка свистка.

Управление

Управление производится с помощью джойстика и некоторых клавиш на клавиатуре.

В кабине паровоза Вы можете перемещать с помощью джойстика стрелку, указывающую на какой-либо орган управления, с помощью джойстика можно и манипулировать им.

Чтобы поднять температуру и давление пара, необходимо открыть крышку топки и подбросить туда уголь (рукоятка джойстика вправо). Скорость движения бронепоезда регулируется рычагом управления давлением пара и тормозами. Чем больше Вы потянете на себя рычаг регулировки давления, тем больше пара будет поступать в цилиндры и тем быстрее будет двигаться бронепоезд. Для остановки нужно перекрыть поступление пара в цилиндры, толкнув рычаг до упора от себя и затормозить, повернув рукоятку тормоза. Приводить тормоз в действие лучше кратковременными импульсами по 2-3 секунды, иначе он быстро выйдет из строя, при закрытой заслонке начинает расти давление пара в котлах и, чтобы удержать его в норме, необходимо стравливать пар с помощью заслонки (STEAM BLOWOFF).

Из кабины паровоза Вы можете перейти к одному из зенитных пулеметов, расположенных спереди и сзади бронепоезда, это необходимо при отражении атак немецких самолетов. Переключение осуществляется нажатием клавиш "1" или "2": возврат в кабину - "3". О налете вражеской авиации Вас проинформирует компьютер, он же выдаст сообщение, если давление пара будет превышать норму или будет слишком низким. Компьютер проинформирует Вас и в том случае, когда запасы угля или воды будут подходить к концу.

Включив "Таблицу состояния" с помощью клавиши "5", можно узнать о состоянии различных систем бронепоезда, из таблицы можно узнать о процентном износе тормозов, котла, брони, и также о количестве оставшегося угля. Выход из таблицы - нажатием клавиши "5".

С помощью клавиши "4" можно вывести на экран карту сети железнодорожных путей запада Франции, на которой будут отмечены все станции, мосты, развилки и также указано положение Вашего бронепоезда. По этой карте Вы узнаете о приближении бронепоезда к станции или мосту, а также спланируете дальнейший путь состава по оптимальному

маршруту. Другие клавиши:

"SPACE" - пауза.

"R" - выход, из игры.

"S" - вкл/вкл звука.

Мост (BRIDGE)

На мосту, захваченном союзниками или отрядами сопротивления (о чем свидетельствует изменение его цвета на карте), Вы можете не останавливаться. Но как только попытаетесь проехать через мост, который контролируется немцами, без остановки, то будете немедленно уничтожены.

Поэтому при подходе к мосту необходимо замедлить скорость до минимума и, когда расстояние до моста сократиться до нуля, о чем Вас проинформирует компьютер, полностью остановить бронепоезд. В этот момент Вы автоматически окажетесь в орудийной башне и должны будете расстрелять немецкие речные суда, которые тоже будут отвечать огнем. Как только все суда будут потоплены, путь станет свободен и можно двигаться дальше.

Станция (STATION)

На станциях, захваченных врагом, можно не останавливаться, но после особенно длинных перегонов или перед ними необходимо заправить баки водой и загрузить уголь, это можно сделать лишь отбив у врага станцию. Для захвата станции нужно действовать аналогично случаю с мостом. После остановки Вы окажетесь на месте пулеметчика и Ваша задача - подавить сопротивление немцев. После захвата станции Вам предложат ознакомиться со сводкой германского командования. Нажав клавишу "Огонь", Вы увидите следующее меню:

- захват следующей станции;
- захватить следующий мост;
- сделать ремонт;
- ничего не делать.

Вы можете отправить соответствующее послание союзникам и получите ответ, в котором будет указано, в какое время произойдет данное событие (текущее игровое время указывается на циферблате в углу экрана).

После этого Вы можете продолжать движение.

Развилка или перекресток (SWITCH)

Повернуть на другие пути можно только при полной остановке бронепоезда на развилке при помощи рычага реверса.

Внимание!

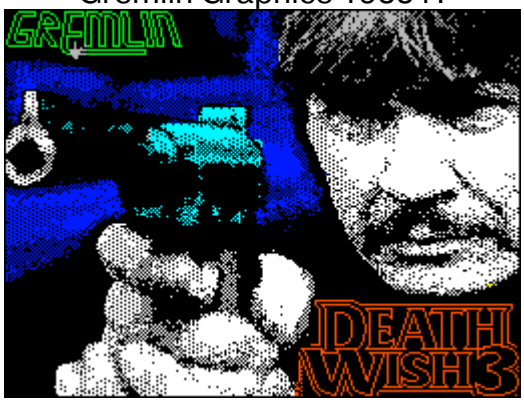
Не трогайте рычаг до полной остановки бронепоезда.

На первых порах будет очень трудно справиться с управлением бронепоездом, но потренировавшись некоторое время, можно приобрести опыт и добиться больших успехов в этой необычной компьютерной игре.

DEATH WISH 3

("Жажда Смерти - 3")

Gremlin Graphics 1988 г.



Эксперт Троекуров В. И. г. Киев.

Фирма GREMLIN GRAPHICS выпустила свою программу на рынок вслед за появлением третей серии всемирно известного кинофильма "Жажда Смерти", главную роль в которой играет очень симпатичный и любимый многими мужественный Чарли Бронсон.

Те, кто не знаком с этим киносериалом, нередко полагают, что где-то существуют программы DEATH WISH 1 и DEATH WISH 2. Но к сожалению это не так. Цифру "три" программа получила от третьей серии фильма.

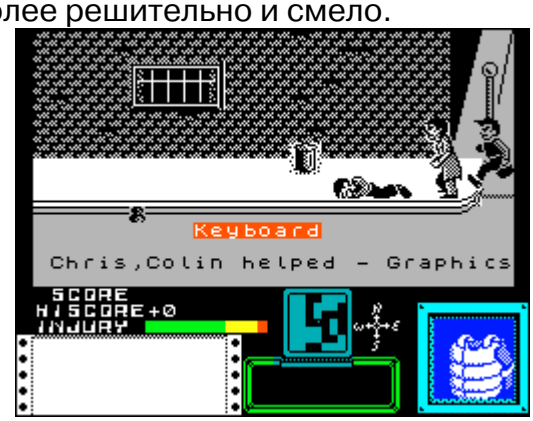
Краткое содержание картины. Скромный, хотя и талантливый архитектор, враг всякого насилия, дрожащими руками берется за оружие в первой серии картины, когда грязные бандиты убивают его жену и тяжко травмируют дочь. Выйдя в одиночку на ночные улицы города, он начинает свою большую и страшную месть всем, кто не дает людям спокойно жить. Постепенно в его действиях начинает проявляться профессионализм. Пролив реки крови, он остается на свободе, поскольку полиция, мягко говоря, смотрит сквозь пальцы на его подвиги во всяком случае работы ей становится с каждым днем все меньше и меньше.

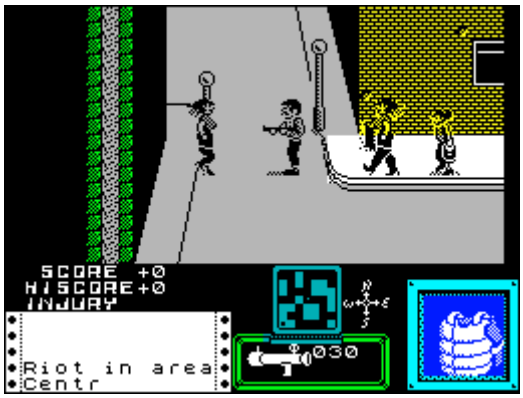
К третьей серии картины он уже сделал несколько безуспешных попыток "завязать", но обстоятельства по-прежнему вынуждают его снова и снова браться за оружие, Теперь он уже большой профессионал с известным по всей стране прозвищем "Мститель". Несколько разрушенных до основания кварталов и полное очищение города от неуправляемых банд - достижение третьей, но не последняя серии картины.

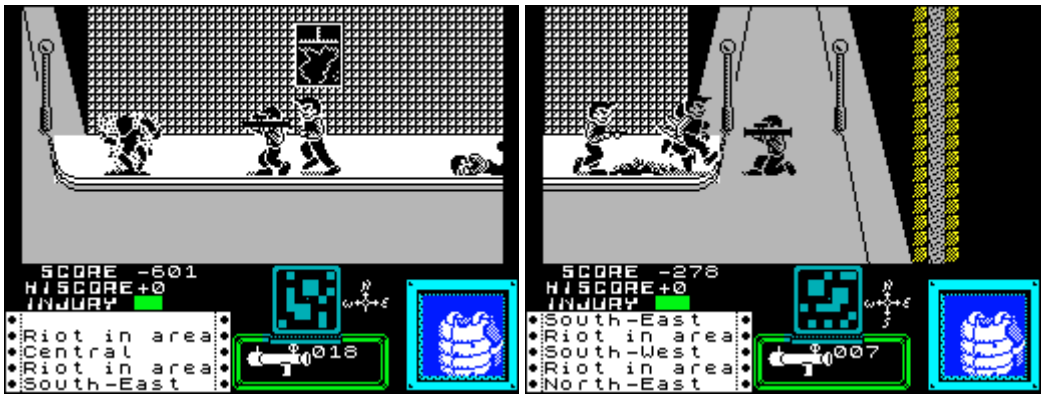
Игра относится к жанру ACTION, но поскольку по ходу игры необходимо собирать оружие, и есть возможность произвольного выбора маршрута, следовательно в ней есть элементы ARCADE/ADVENTURE.

Сюжет: полиция не может освободить город от бандитов, ведущих между собой борьбу за власть в городе. Вооруженные банды устраивают стычки с полицией, во время которых проливается кровь мирных жителей.

Вы в качестве главного героя должны навести порядок в городе, а так как вы свободны и независимы, в отличие от полиции, которая связана законами и уставами, то можете действовать более решительно и смело.







После загрузки программы нажмите "BREAK". Вы увидите нашего главного героя с винтовкой - это Стив, которым Вы управляете. Теперь можно начинать игру.

В нижней части экрана выводятся:

- выбранный вид оружия и боезапас;
- информационное табло, где приблизительно указывается местонахождение главарей банд (их пятеро и находятся они в пяти районах города: NORTH - WEST, NORTH - EAST, SOUTH - WEST, SOUTH - EAST, CENTRAL);
- состояние бронежилета (повреждения от выстрелов, возможность замены на новый);
- карта участка города показывает ваше местоположение (с указанием направлений: север, юг, запад, восток. Синий цвет карты - Вы настроены на поиск оружия, желтый - поиск главарей банд);
- SCORE - заработанные Вами очки;
- HI SCORE - высший результат предыдущих игр;
- INJURY - состояние Вашего здоровья (если показалась зеленая полоса, спрячьтесь в укромном месте и отдыхайте до тех пор пока полоска не исчезнет);

Итак начинайте игру. На улицах и в домах на Вас нападают бандиты, вооруженные винтовками или дубинками, они не церемонятся - нападают и убивают на месте. Иногда будут пробегать полицейские и стрелять по бандитам, но от них мало пользы и рассчитывать Вам надо только на свои силы.

Оружие. У Вас имеется:

- пистолет;
- автомат;
- карабин;
- бронежилет.

Оружие и бронежилеты находятся в разных частях города - в домах, куда Вы можете зайти и пополнить боезапас или сменить вышедший из строя бронежилет.

Смена оружия производится нажатием клавиши "C". Нажатие клавиши "M" поможет Вам найти главарей банд, повторное нажатие "M" поможет найти оружие, боеприпасы, бронежилет. Пауза - клавиша "H".

Главари банд также находятся в домах. Для входа в дверь служит клавиша ENTER. Когда Вы найдете главаря, его надо расстрелять. Первым выстрелом его не убить, но зрелище того, как рассыпается стол, за которым он сидит, доставит Вам большое удовольствие.

Крайне нежелательно стрелять в простых мирных граждан - за это Вас штрафуют, но самое большее преступление, которое вы можете совершить - убийство полицейского. За ним последует солидный штраф в очках и необходимость спасаться не только от бандитов, но и от полиции, а это уже конец.

FORUM

Проблемы ELITE

Мы по-прежнему получаем сотни писем по программе ELITE, во сейчас информация, содержащаяся в них. как правило повторяется, какие темы в основном волнуют читателей:

1. Беспричинные захваты корабля на пиратских станциях, очень много жалоб по этому поводу. После приземления сообщают, что ваш корабль зажвачен пиратами и действуют они безжалостно. Вам приколите я начинать игру сначала. в качестве конкретного адреса приведем сообщение Тихомирова в. в. и Митрохина в. а. из г. чернонорска (крым) о том. что так ведет себя стандия на планете LAZASO в га лактике N5.

2. по-прежнему иного сообщений о чудачествах в версии, взломанной Родионовым. здесь и галактика N47 и галактика к 1 без планеты LAVE и головокружительные суммы кредов и необъятные возможности вооружения в оеюем не знаем, хочется ли вам в этот бред играть, но писать о нем нам не хочется.

3. после того, как мы в ю-и номере прошлого года опубликовали изображения веек документирован как кораблей (как оказалось не зря!) пилоты начали настоящую окоту за нло. Сообщений много и как правило они имеет одиночный характер, но в десятках писем сообщается одно и то же. Есть корабль больших размеров, бесформенный, похожий на астероид, несущий на себе "Крейты" и "Саид-уиндеры". сам первым не нападает и. выпустив истребители, старается уйти от боя. Если его поразить, сбрасывает контейнеры. Факт существования такого недокументированного корабля можно считать установленным точно. Его поведение описывают десятки людей одними и теми же словами. Это кандидат на то. чтобы считаться "Космической платформой'. Условно назовем его пока STRAHGER ("странник").

Рисунок взят из письма н. Ушакова (г.Ангарск) - см. рис.1.

4 есть разночтения по поводу подзарядки топливом от звезды, мы не зря поднимали эту кажущуюся тривиальной проблему. 9ох делают это не задумываясь, а юх не могут сделать этого вообве. у них

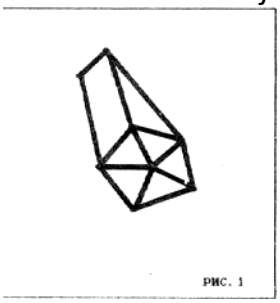


РИС. 1

перегревается корабль задолго до включения надписи 'Fuel SC00PS on" или просто температура не растет, но и заряда нет.

ножет быть стоит прислушаться к мнению нияайлова с. ю. (г. Саратов). который пишет,что "совка кк ill" имеет нерегулярную геонетрическую форму и поэтому есть раз ниua. каким бокок Вы ориентире ваны к звезде, корабль, поверну тый плоскостью, нагревается го раздо сильнее, чем корабль, развернутый торцем. "

Чтобы окончательно решить этот вопрос. Пусть нам напишут только те. у кого заправка никак не идет и сообшат о тон. какой версией игры они пользуются. Может быть выяснится, что она у всех одна и та же.

в последнем выпуске (11-12) прошлого года мы сообщили о тон. что пилоты начали исследования юг-байтного блока состояния, который отгружается, когда вы сохраняете программу на ленте. Сегодня Радзевич а. а. из г. Нальчика дает дополнения к ранее неустановленным байтам, 15 - Ваш рейтинг. 16. .. 1в - выполненные миссии. 41 - грузоподъемность (тонн). 61... бб - расположение звезд в галактике.

67 ?

68 - координата v корабля.

69 - ?

70 - координата x корабля.

Интересное исследование провел пилот класса ELITE из г. Екатеринбурга д. палтусов.

Во-первых, он принудил свою версию программы работать с джойстиком, хоть она и очень не хотела:

роке 29646.191: роке J9649.2j6: роке 2905в. о: роке 29659.0 (у него версия, помеченная Джеком О'лантернон).

во-вторых, он изучил машинный код программы и ему удалось уста-новить кое-что интересное. как оказалось, программа ведет внутри себя собственный подсчет очков.

за каждые 256 очков выдается сообщение RIGHT OK COHHAMDER.

а вот как оценивается ваша боевая активность при уничтожении кораблей:

THARGOH - 4 балла. FER-DE-LAHCE - 3 балла. ASP HK II - 3 балла FYTHOH. ADDER-2. COBRA MK III,

SIDEVIHDE& и 'отшельник- - по I

баллу.

контейнер, астероид и VIPER -не дают очков.

Установлено соответствие и между набранной вами сунной баллов и Вашим боевым рейтингом:

рейтинг очки II

HARMLESS o |

MOSTLY HARMLESS 9]

P00R 17 I!

AVERAGE 33 i

ABOVE AVERAGE 65

COMPETENT 129

DANGEROUS Tf>8

DEADLY ZS16

KL1TE b&'^b

кстати, это ответ тем читать лян, которые недоумевают почему у них останавливается нарашинанио рейтинга после достижения COMPETENT.

исследовав карты галактик, наш читатель обнаружил и левиди мые звезды, как и предполагалось ранее, это двойные звезды. получаемые наложением по OVER i. AFJ тору удалось просчитать их коли 1 чество и расположение. I

галактика звезды I! ?. DIИAATHA - Jaxек 1 се н

4 OU7AARAR -RIUSMUQU i

5 EDXEHERE-GEDIESOU н LAZAZO - ZAKHZA I TE2ABI - DIVKKS 1 ORLAED TET1RJ н

7 DICELASO 2AANXEGE I! ERVEAK - SOiJISOAN н

в CKCKKS ESUSALE II

есть еще одна пара в 8 ой га н

лактико. но найти ее пока нашему II

корреспонденту не удалось. !!

Итоги конкурса на лучший технологически развитый маршрут.

читатели 1991 года поинят. что в N ю ZX-ревю мы предложили пилотам игры "ELITE" составить для любой галактики план наиболее технологически развитого маршрута из го пунктов, сумма технологи ческих уровней в нем должна быть максимальной, сегодня мы можем подвести итоги и объявить пять победителей.

всего на конкурс поступило около 30 маршрутов, как правило, пилоты, приславшие их. имеют рей тинг "DEADLY" или 'ELITE-. Да это и не удивительно. маловероятно. чтобы начинающий пилот смог обяе теть восемь галактик и найти наиболее интересные с этой точки зрения планеты.

как оказалось, удача в поисках сопутствовала тем. кто удачно определился з выборе галактики для прокладки своего маршрута, наиболее широко были представлены галактики HI, г. з, 7. в и вот каковы средние результаты в этих галактиках:

HI - 19S

кг - 219
нз - 206
н7 - 210
N8 - 219

нет ничего удивительного, что те пилоты, которые выбрали для исследования галактику н7 оказались в наиболее выигрышном поло-жении.

вот первые 5 маршрутов-победителей (все они проложены в седьмой галактике).

1.

Пилот: Марченко а. порт приписки: г. новая Каховка

(Херсонской области). Боевой рейтинг: "ELITE" Боевой стаж: 2. 5 нес. Маршрут:

BIOHBIIH йАЙАиЗ - CERIANOH -KKAKIHES - ESTIRI - OHAHO RRA - XEUSREOR - пакакеке - ZARAUSXE -AGEBI - ESZAUSVE - AHGERIRI -&UGEZA - ESDITI DIUSACE -ESRIXEA» - TIREGEES ORESATRA QUDIOR - ATESLETE. суммарный технологический уровень маршрута Ы балл.

2.

Пилот: минеев а. в. порт приписки: Владивосток Боевой рейтинг: not supplied Боевой стаж: not supplied Каршрут:

BIOHBJIN - MARAUS - CERIAHOH -ENARINES - ESTIRI – OHAHO RRA -XEUSREOR - MARARERE - ZARAUSXE -XEVEXEAN - AGEBI - ESZAUSVE -QULAAOH - AHGERIRI - QUGEZA - ESDITI - AEDEDLE - DIUSACE -TIREGEES - eSRIXEAR

суммарный технологический уровень маршрута - 248 баллов.

3.

пилот: ветлянский а. в. порт приписки: Анадырь Коевой рейтинг: not supplied Боевой стаж: not supplied маршрут:

BIOHBIIH HASAUS - CERIAHOH -EKAitIHES - ESTIRI OHAHO RRA - XEUSREOR - HARARERE - USOUEH -ZARAUSXE AGEBI ESZAUSVE -AHGERIRI - QULAAOH - ISARE -ESDITI - DIUSACE - AEDEDLE -ESRIXEAR - TIREGEES

суммарный технологический уровень маршрута - 248 баллов.

4.

пилот: Сытиик в. а. порт приписки: г. Угледар, донешсои обл.

Боевой рейтинг: not supplied Боевой стаж: 3. 5 года Маршрут:

BIOHBIIH - HARAUS - CEKIAHOH -EHAEIHES OHAHO RRA - XEUSREOR -HARARERE - USQUEH - ZARAUSXE -AGEBI - ISARE QULAAOH -ESZAUSVE - AHGERIRI - QUGEZA -ESDITI - DIUSACE - AEDEDLE -ESRIXEAR - TJREGEES

Суммарный технологический уровень маршрута - 246 баллов.

5,

пилот; старцев а. г. порт приписки: г. миасс.

Челябинской обл. нзршрут:

TEABI - EREMRA - XEGEARER -BIOHBIIH - HARAUS - CERIAHOH EHARIHES - ESTIRI – OHAHO RRA -XEUSREOR - HARARERE - USQUEH -ZARAUSXE - ISEMES - ISARE -QUGEZA - QULAAOH - AGEBI -AHGERIRI - ESZAUSVE

суммарный технологический уровень маршрута - 242 балла.

Как Вы можете убедиться, все маршруты-победители имеют немало обиих пунктов, таким образок, можно считать, что наиболее технологически развитая область в программе ELITE установлена с достаточной точностью.

в соответствии с условиями нашего конкурса все пилоты прислали вместе с маршрутами выписки из бортжурнала. в которых указано какие товары целесообразно покупать на каждой из планет, и вот. что ни установили:

посетив юо плавет, участники конкурса 29 раз проголосовали за приобретение компьютеров и 19 раз - за меха. Прочие товары уступают но своей эффективности. вывод очень прост: хотите иметь неха -осваивайте компьютеры. Эта мысль может иметь далеко идущие последствия.

победителям мы высылаем ксерокопию повести "THE DARK WHEEL'. написанной по мотивам программы "ELITE".

теперь новое предложение:

Мы предлагаем проложить наиболее рискованный маршрут, Условия те же, в любой галактике облететь 20 планет, не побывав на одной планете дважды и подсчитать уровень опасности этого маршрута, давайте условимся, что за анархическую планету начисляется 5 баллов, за феодальную - 4 балла, а за лобую другую - i балл.

точно так хе. как и в прошлый раз. просим прислать выписку из бортжурнала. в которой указано, какие товары наиболее целесообразно покупать в каждой из точек вашего маршрута, чтобы иметь в

оче редном пункте наксинад ьную прибыль. Эти данные будут свидетельством того, что вы действительно пролетели по маршруту, а не взяли его из Веегаллактичеекого справочника.

Кроме того, интересно узнать, а на чей же делают бизнес в самых злачных уголках вселенной.

задача несравненно более трудная, по сравнению с той, что была поставлена в прошлый раз. мы знаем, что у этих планет вас будут подстерегать пиратские Флотилии, но зато и ценность ваших наблюдений будет максимальной. Начинающим пилотам можно будет выдать ре коне ндашш о том, в какие районы им не следует показываться, а опытный бондам высокого класса будет ясно, где наиболее быстро можно поднять свой рейтинг.

для документальности Ваших отчетов Вы можете сообщить свой боевой рейтинг и стаж полетов. Будет приятно также напечатать о количестве одержанных вами побед во время прохождения маршрута.

призы те же - 5 ксерокопия повести "DARE VHEEL" и публикапия отчетов в майском/июньском выпуске.

отчеты принимаются до 1.05.92 по почтовому штемпелю даты отправки.

POKES

л. палтусов, который только что поделился своими результатами исследования ELITE активно исследует и другие программы, вот список некоторых POKES, которые ему удалось отыскать самостоятельно:

ТиТАНКНАНии 277в3,0 HOTOS 48241.0 LODERUHHER 354Z7,24: 3542б,о DOWH TO EARTH 40142.19S SAHXXOH 36584.0 KNIGHT LORE 53567,0: 50205.0: 50206,0: 50207.0: SPELLBOUND 27036.R (в начале)

BATTY 48437.167

CHROHOS 53407.H

BOULDER 4 30960.H

COHAHDO 27652-27657.0

BOULDER 1 3I007.0: 31008.0

310O9.0 HETAL ARMY 4&535.0: 48559.0

(энергия]

42198.0: 48700, 107

(жизни)

CRAKOUT 46565,0: 61534,0 H. o. H. A. D. 4O167.0 REMEGADE 410чт.36 ACADEHY 31378.H: 31386,H

31249.H: 31305.H (н-оснастка корабля) EQUIHOX 41914.0 HYSTERIA 445B»,201 L. HIHJA 2 36579,175:36578.0 GUHFRIGHT 49233,5»: 49234.1

49E35.0: 49236.0

ASTEHEJC 43516,0

AIR FORCE II 519O4.0

CTBEBHOID 39402,0

CVBEBNOID Z 36197.0

FRED 31171.0

IMPACT 5450O.183

два слова о тон. как ищутся адреса для POKES на примере программы н. о. н. A. D

зная, что в программе в искод-нон состоянии игрок имеет 3 попытки, наш читатель предположил. что где-то есть ячейка памяти, в которой хранится это число, иожно также

предположить, что засылается оно туда в начале работы программы через регистр а. это, конечно не единственный способ за-сылки числа в память, но он достаточно удобен и широко распространен.

просмотрев с помопьс дисассемблера программу, он сразу отбросил те области, в которых хранится графика. спрзйты, тексты и т. п. . сосредоточившись на нашивном коде. (Как можно всего за несколько минут прикинуть где что в программе находится, "инфорком" писал во г он тоне трехтомника по программированию в машинном ко де.)

Такой беглый просмотр позволяет выделить для более подробного исследования область длиной всего в несколько к. в этой области ищется коиада АССЕМБЛЕРа

LD a. з.

Это занимает еще несколько минут, если таких точек несколько, их надо будет все испытать, что делается следующим образом.

Рассмотрим фрагмент программы:

```
LD a. 3
LD (34566).a
LD (34450I.a
LD HL,27304
LD (34584),HL
```

Итак, есть подозрение, что в ячейке 31536 или 34460 организована переменная, в которой хранится количество попыток.

Давайте проверим ячейку 34586.

с помошыо поисковых возможностей дисАССШБЛЕРа найден где еще упоминается этот адрес и досмотрим, что там записано. Обычно это выглядит так:

```
LD a.(34586)
DEC A
LD (34586).a
```

команда DEC а. уменьшающая значение в аккумуляторе на единицу. - прекрасный кандидат на то. что мы ишем. ведь после гибели героя количество попыток уменьшается ва одну.

проверим это. заменим DEC а на команду нор (нет операции - ее код =о>. в запустим программу после этой переделки. Если мы попа-

ди правильво. ваш герой теперь бессмертен.

а вот как ищутся пароли для игры, возьмем программу SABOTAGE. пройдя первый уровень удалое ь установить пароль второго < его программа выдала сама) - "BUHBLE

нее 2".

Теперь найдем место в программе, где хранится эта фраза, ножно даже не пользоваться дислссЕНБЛЕ ром. а сделать это из БЕИСИЕа.

Код буквы "в" равен 66. а код буквы "U" - равен 85,

```
10 FOR i=25O00 то &5535: IF PEEE 1 - 66 AMD реек (1+1) = 85 тнен PRINT 1 20 NEXT 1
```

когда будет найдено место в программе, в котором имеется сообиение "BU... ". компьютер выдаст вам его адрес, проверьте близлежащие адреса в них может быть тоже что-либо интересное, так удалось установить пароли прочих уровней:

2) BUHBLE BEE г.

3) HONORARIUM.

4) рнененонон.

5) OHOMASTICS.

6) SALMAGUNDI.

7) PSEUDOHIHOUS. в) ONOHATOPED1A. мы благодарим д. Палтусова за интересную информацию и вынуждены только пожалеть, что в ELITE COMPETITION он не занял призового места, он нашел великолепный технологически развитый маршрут для галактики HI, но не догадался исследовать галактику н7. как это сделали победители.

Кстати, о паролях. Наш читатель ЖУкович н. н. из г. Усолье Сибирское очень просит помочь ему найти пароль 8-го уровня в игре IMPACT. Может быть кону-то удалось его пройти? для тех же, кто кочет попробовать свои силы в этой увлекател ьной игре, он

сообщает пароли прочих уровней.

в игре 9 уровней по ю экранов, экран 1 - пароль не требуется, экран 11 - EGGS экран Jl - CHIP

Экран 31 - LEAD

экран 41 - TICK экран 51 - CASE экран 61 FACE экран 71 ????

экран 81 - USER - пользовательский.

наш читатель из г. Смоленска, коновалов а. в. тоже активно работает с POKES, к сожалению, он отмучает, что некоторые POKES из тех. что были опубликованы в ню у него не пошли, в то же время, известно, что по стране ходят многочисленные версии даже для одной и той же игру. поэтому для

тех, у кого не пошли какие-то POKES, оя предлагает воспользоваться прилагаемым ниже набором.

все прилагаемые POKES вставляются после последнего LOAD", а не между USR.

GABEOVEF1 39334.0 (жизнм)

32417.0 (граната)

HORACE ало

SPIDE8S 27680.60 MAG-MAX 58472.60 HUTAHT HOHTY 55761.60

56483.183

CYBERHOID- 1 39403.0 V1XEH-II1 41433.o FIREBIRDS 27235,0 XEVIOUS 53592. п BLADE WARRIOR 37161.0

39490, 60

KARNOV 25620.o GAHEOVER 2 38704. 0 (жизни)

згзва.о (гранаты) GOKZZALEZZ 2 35749.0 (жизни)

48056, 0 (оружие) ASTEHEX 43517,52 GONZZALEZZ 1 370в5. 0 METAL ARMY 36897.0 (жизни)

42650.o (JHPPPHH) ACTION FORCE 2 47874.IS2 KR10H 45004,0 (жизни)

14486.o (ГОРЮЧее) GLUG GLUG 34)39.o PSSST 24984, 0 ROGUE TR00FFJ; 309ill. 0 EQUINOX (для 39858. 52:39659. tfli' версии ROBV'36) (жизни)

48691.o:ч8762,o

(горючее и лазер) STAIRLESS STEEL 46957.60 UNDER WURI.IJE ь9376, 0 RICK DANGEROUS 55460.0 (жи:шь)

61045.0 (гранаты)

&095ч.о (патроны)

CAVELOH ч72ъо.о

K-TYPE 30873.o

UAB-DARE 2/1 46459,0

VIGILAHTE 46735.60

кокотош WILF 43742.0

PhTER PACK RAT 27243. 1oo

PEHTAGRAK 49917.0

TH ANDEKBLADE 33199.0

33145,0

THOR 37550.19b

DAK DARE 2/2 51797.0

HOTOS 42241.0

CHICAGO 6O2&4.0

FROST BYTE 30992. 183 (жизнь)

28237.183 (время) RUFF * REDD? 33567.0 PROJECT FUTURE 29332.0 JETHAN 36965.0

комментарии -ИНФОРКОМа$.

Дорогие друзья? то. что игры. имеющие хождение в стране, уже давно не является цервоэдавкы ни. Вам хорошо известно, сначала вх ктрочат в Голландии* потом че-рез фрг это поступает в польшу. где дело поставлено не хуже, чем у нас. и только потом мутными ручейками программы поступают на расправу к нашим специалистам.

вам это известно очень хорошо.

кроне того, для различных POKES существует и разная техно-логия их введения, что тоже не-маловажно и не всегда правильно выполняется. Радикальных рецептов быть не может, кроме одного: учиться, учиться и учиться, как завешал сэр клаяв Синклер, а в англии, как Вы понимаете, звание сэра и титул лорда простому инженеру за олни красивые глаза не (дают. J осваивайте машинный код. Изучайте опыт и приемы тех. кто ! это уже освоил. развивайте кобственные приемы. в принаипе, 1на страницах "ZX-PEHO" мы даем достаточно информации, чтобы было с чего начать. Как Вы увидите, в этом году в "ZX-PEBO-92'.

будет немного больше материалов для тек. кто осваивает машинный кол,

Только освоив технологию вы почувствуете себя хозяином положения и сможете воспользоваться чьим-то роке с полным знанием дела или адаптировать его под свою конкретную программу.

i:

Вопросы совместимости.

* на продолжаем рассматривать 1|вопросы совместимости отечествен-1>1ых моделей 1к с Фирменным прог-Црамнным обеспечением, наши чита-ители уже встречались с разработ Вкаии в этом направлении, ведуиси-Внися Нолубарьевык Сергеем викто-Ировичем из г. йошкар Ола

II Сегодня речь идет о наведении я порядка <; дешифрацией портов 1| ввода/вывода в версии зонова i| (1 5 корпусный "Ленинград-1 i для Обеспечения полной совместимости (It; оригиналом.

J! Вот последовательность дора -ьотки. для этого необходимо уста -ноиить я<1 свободном месте платы по одной микросхеме к5 55лл1 и к'-н1ьли1 и лалее выполнить в схеме следующие изменения:

1) отключить сигнал IQWK от

ньшод<э 9 DUJ9 <к5^5тму Все ОбО-

:m<j одним по принципиальной схеме "Ленинград, i" i и собрать на одном нз элементов къь5лл1 следуиаую

с, к е ну ;

после это;-о запись в порт вывода будет происходить только по четным адресам, что исключает мерцающе и самопроизвольное изненение цвета бордюра.

Небольшое отступление перед последующими доработками:

"Ленинград-1". по-видимому. разрабатывался под самодельный джойстик с нормально-замкнутыми контактами, наиболее же распространенны»! являются джойстики "Atari"-типа, у которых контакты нормально разомкнуты и замыкаются при отклонении рукоятки, часть последующих передедок будет направлена на обеспечение совместимости именно с няни.

2) РеЗИСТОРЫ R21...Rg5 сопротивлением 15 ком, подключенные к линиям джойстика DVO... DV4 соответственно, заменить на нлт-о. IES с сопротивлением порядка

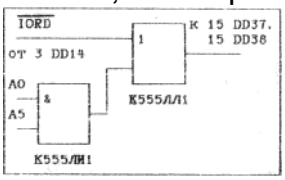
660 он. . . 1. 5 кон. ОбШУЮ точку их

соединения отключить от шины • + 5V" и соединить ее с "землей".

3) свободные входы отз? (выводы 6.10,13) соединить с выводом а этой же микросхемы.

после выполнения этих доработок в принципе уже можно включить компьютер и оценить результаты. команда PRINT IN 31 должна при отпущенной джойстике печатать о. а при отклонении рукоятки или нажатии кнопки - некоторое число, зависящее о положения джойстика, запись любого числа по любому нечетному адресу OUT не должна изменить цвет бордюра. Однако, если вы хотите довести джойстик до полного Кемпстон-стандарта, а заодно и освободить часть нечетных портов ввода, необходимо сделать еще одну доработку:

4) отсоедините сигнал IORD. поступавший с вывода з DD14 (к555лл1) от вывода 15 DD37 и DD3S, и соберите следующую схему его дополнительной доработки:



Теперь считывание из сортов будет производиться так:

ао о и а5-1 - клавиатура; ао-0 и а5^о - клавиатура;

т. с. клавиатура читается по любому четному адресу корта ао ! и аь-о кемпстон-дж--к. AO^I и Ab^i несуществующий порт, т. е. все нечетные порты с адресами, большими, чем 3S свободны.

Практика показывает, что при выполнении доработок 1. , . 3 большинство проблем исчезает, дора-ботка 1 не является обязательной (в игровых программах обычно эффекта не дает) и применяется во-первых, чтобы обеспечить соответствие стандарту и. во -вторых. чтобы освободить часть адресного пространства, например для уста новки "ZX-LPffIBT III".

Соответственно, если необходимости в ней нет. микросхему к555/ш1 можно не устанавливать. оставшиеся свободными элементы дополнительных ндс можно использовать, например. при подстыковке интерфейса "BETA-DISK".

Если рассмотреть доработку. преддагавяюся в ZX-ревю в прошлом году (стр. 157. рис. 7) , то

можно понять, что в принципе она делает то же самое, но при этом со старших 3 битов порта 31 продолжает считываться единила, что в принципе может в некоторых случаях приводить к нарушению совместимости и в некоторых играх и приводит.

Теперь несколько слов об отмечавшемся замедлении работы игровых программ на "Ленинграде-1"

(см. 2х-ревю-91, с. 197).

прежде всего, примем ээ аксиому следующее:

а) схема достроена так. чтобы использовать кварпы от 13,0 до 14.5 МГц. тактовая частота процессора получается при делении частоты генератора на 4, т. е. это 3.5 МГц при частоте кварца 14 МГи. Поэтому первым делон проверьте кварц, который стоит в вашей машина

б) Конфликт дисплея и процессора to которой писали на странице ьг "zx-ревю-91") может происходить по любому адресу озу, поскольку у "Ленинграда-1" сплошное поле памяти 46к. в этом случае процес сор приостанавливается по сигналу WAIT, формируемому на триггере D 9. г. Исходныни для Формирования служат служат сигна лы и1 процессора и CSRAH (выборка ОЗУ поступает на вывод 1 регистра К555ВРЕ2). Однако, в некоторых схемах и рекомендациях до наладке (соответственно и в платах) вместо сигнала CSRAH требуют исполь зовать сигнал MREQ. последствия этого - замедление работы даже с ПЗУ (хотя на некоторых экземоля рах компьютера это, возможно- и необходимо;.

Методы устранения несложны.

а) Замените кварц на 14.0 или 14,5 нгц. Чтобы при этом не нару шалась синхронизация TV, внесите изменения в схему включения счет чика ш (к555ие7) согласно следу-юшей та бдит;;

———	выводы D4 соединить с:	
——— — частота .нгц	"•ьвояьт"	"ОбШИЙ"
13. 0	10. 15	1, 9. 14
13. 5	10	1.9, 14. 15
14. 0	1. 15	10, 9. 14
14.5	1	10,9. 1'i. 15

Если после замены кварца обнаружатся сбои озу. придется аоме нять микросхемы на более быстро девствуюиие или попытаться под строить временные диаграммы сиг-

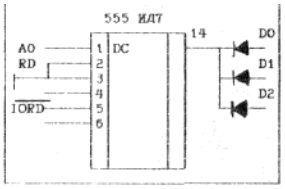
налов RAS и CAS.

6) Решение второй проблемы вытекает из самой проблемы, оона-ко, если после замены на Формирователе HREQ на CSRAH обнаружатся сбои озу. придется вернуть все на место.

в любом случае рекомендуется после проведения доработок выполнить длительную серию тестов озу. чтобы Убедиться в надежной работе конпыотера.

Все предлагаемые доработки проверены автором лично на практике, во избежание появления сбоев лучше применять в компьютере "Ленинград-1" только серию к555 (к1533). не заменяя ее на к531 и к155 - тогда наладка почти не требуется.

для тех. кто работает с Краснодарским и харьковским вариантами компьютера, впрочем как и с "Ленинград-1". наш читатель из Харькова в. и. бойко предлагает несложную доработку, обеспечивающую работу таких программ, как •ARKAHQID". "DUET" и неКОТОРЫХ других.



На ножку б микросхемы подается старший байт адреса, формируемый схемой генерации адреса ULA. все остальные сигналы можно взять непосредственно с процессора.

Просьба о помощи,

в Нежгакольжж комбинате г. Кайли Сай (Кыргызстан) установле на локальная сеть на базе "Соек трунов", изготовлении! в Ташкенте. Проблема с эксплуатацией программ в сети. Программы есть, но Нпо сети не идут. Если у кого есть опыт зкснлуатсшии сети, просьба отозваться. "инфорком" рассматривает вопросы обеспечения работы школ, как самые приоритетные.

Контактный адрес: 7154го. Кыргызстан, джадал-Абадская область, г. яаиаи-саи, ул. наяковского, 2, кв. Jl,

курику в. н.

Сам "инфорком" располагает всей необходимой информацией по тому, как организуется локальная сеть для Фирменных машин на базе интерфейса локальной сети ZX-INTERFACE I. Известна система команд. системные переиевяые. коды-перехвата, но мы со&еряенно не знаем, на каких интерфейсах делают сети у нас и какими командами они задействуетея.

Примем к печати статьи- на давНУЮ тему.

и снова SHERLOCK.

Нашим читателям уже известны проработки эксперта Ескевича а. а. из г. Новосибирска, как студента-филолога, его конечно интересуют текстовые игры (жанр ADVENTURE) и он активно взялся за работу с программой SHERLOCK, но столкнулся с проблемой.

Узнав от Уотсона о том. что в Лизерхэде (LEATHERHEAD) произошло убийство двух человек, он решил направить Холмса на место преступления, проблема в том. что ни один извозчик не желает его понимать, он побывал на всех четырех вокзалах и на прилегающих к ним улицам (Aidereate street. Bishops Road, BacKingbam Palace и

к1п83 сгозз), но ни один извозчик ниоткуда не желает везти его в Лизерхэд.

Мы с большем удовольствием принимаем эту жалобу на безобразную работу лондонских извозчиков и должны заметить, что у нас в Носкве тоже очень часто бывает, что таксисты не хотят Вас везти ниоткуда и никуда, не знаеи, как обстоят дела в Новосибирске, с другой стороны, есть сведения о том, что некоторым удается как-то договориться с водителем, возможно, есть какие-то филологические тонкости в обвении пасаажира и водителя нам яеведомые.

понробуите применить такой же подход и к лондонским кэбби. одним словом: "Вам нужно найти необходимые слова". Вот они:

выйдя из дома. Холмс спешит на утренний поезд в Лизерхзд, который отправляется в

9:15. Для этого ему нужно нанять кэб:

1. HAIL CAB

г. CLJHB INTO CAB

3. SAY TO CABBY "GO TO ..."

вот и все. Этин же методом Вы будете нанимать кэб и в других подобных ситуациях.

На платформе Холмс встретит инспектора лестренда ... а далее желаем Вам успеха.

а вот, что швиет антон сквор нов из с. Петербурга. "Неня сразу заинтересовали ваши статьи об ад&еитюрнык играх. Пне они во многом помогли, правильно, что Вы развиваете интерес к этому жанру.

я. вравда, еще не очень олытен и хотел бы переписываться с кои ннбудь. кто любит эти игры. Проб-яеиа в тон. что на навей "толчке" их не достать, и я пока довольствуюсь двумя программами кшвйп-туюе и The Hobbit. "

Уважаемый антон, развивая это направление, мы сознательно шли против течения. против моды. против законов местных "толчков". Да, сегодня энтузадстов этого ханра пока мало. Но есть мировой опыт, и он гласит однозначно -через д.ва-три года общения с компьютером основными становятся игры трех хаНРвв - ACVKHTU8E,

STRATEGY. надасенент. Так что Не

беспокойтесь, если на Вашем "толчке" есть умные предприниматели, они это поймут, перестроятся и в течение года - двух ситуация резко изнежится.

Наша задача - подготовить этот поворот, а Вы правильно делаете, что осваиваете это направление, когда придет его время, будете уже иметь нужный боевой (а может быть и деловой) опыт.

для любителей -умных- игр, желающих связаться с Антоном: 195огт. с. -Петербург, среднеох-тинский проспект, д. 2 "а", кв. 1эг. _____

Внимание: Будьте осторожны. I

наш читатель из г. Харькова хоминич р. в, сообщает о неприятном моменте. который поджидает многих любителей аркадных адвен-тюр, начавших игру в программу BLACK RIDER фирмы TOPOSOFT. 19йв.

прежде чем начать с ней работать, желательно проверить ее на полноту состава Файлов.

а вот в чем суть.

Вы являетесь капитаном пиратского корабля, корабль находится на стоянке в бухте и ремонтирует ся. с каждым днем становится все труднее управлять разлагавшейся командой. нэтросы пьют ром и готовы разбежаться.

Ваша первая задача - сохранить команду. Разрушить скодной трап можно выстрелом из пупки, но надо найти фитиль (Факел).

Поиск ведется в многочисленных сундуках с сокровишами. расположенных в различных повешениях корабля. Чтобы не тратить время да возню с замками, капитан открыва-ет их не церемонясь выстрелом из пистолета.

Среди прочего барахла в этих сундуках Вы найдете и полезные для себя веши.

Все время вам придется отбиваться от своих подгулявших ПОДЧИнеиных. Интересна система подкрепления энергии. Подкрепиться можно ромом, но будьте осторожны, не переберите.

когда вы обышете все сундуки, на самой нижней палубе вас будет ждать сюрприз. Старый красный сундук. который до этого никак не открывался. Вдруг окажется открытым, а рядом с ним лежит удавсаяь-ная карта острова сокровищ. Вы бросаетесь к ней и. - . программа просит загрузить следующий уровень, что очень неприятно, если у вас его нет, а именно в таком виде программа и нашла широкое распространение.

Тех счастливых джентльменов удачи, которые владеют полной игрой, мы просим прислать список ее файлов с указанием длиш; и названии. дабы многочисленные поклонники аркадных адвентюр не становились жертвами столь бездушного обращения.

"РЕГИСТРАТУРА' - система многоцелевого назначения.

система относится к РЗЗРЗДУ информационно-поисковых и справочных систем, исполняемых и настраиваемых

-под заказчика". Данная система полет удовлетворить большинство потребностей организаций в удобном программном средстве для хранения и обработки текущей информации.

I. НАЗНАЧЕНИЕ СНСТЕНS

система предназначена для ведения учета и проведения обработки тобой информации, необходимой на вашей предприятии ев вашем учреждении). Например:

- регистрация вводящих и исходящих документов;
- регистрация входящих и исходящих товаров;
- регистрация пациентов в лечебном учреждении;
- регистрация клиентуры и заказчиков
- и многое, ннорое другое.

система может удовлетворить потребности отдела кадров, планово- Финансового отпела, отдела соикультбы-та. различных административных, нуюшноалъшк и хозяйственных служб и т. п. . кроме бухгалтерии.

II. ВОЗМОЖНОСТИ СИСТЕМЫ

Система позволяет:

- вводить инфорнашш в любое заданное количество полей;
- просматривать информацию в избранном Формате;
- вносить изменения:
- сортировать информацию по любому полю или по любой их совокупности;
- производить поиск по заданному критерию или по любой их совокупности;
- произвольно ио каждому критерию задавать ключевое соотношение на поиск;
- получать статистическую информацию (производить подсчет записей, удовлетворяющих любым заданным критериям при любых заданных соотношениях);
- распечатывать всю или избранную информацию на принтере в избранном составе, избранный шрифтом на листах избранного формата;
- делить избранные файлы на части в автоматическом или ручном режиме;
- объединять файлы:
- …. и многое другое.

Практически все операции, включая выбор формата просмотра, выбор критериев на поиск и сортировку. выбор ключевых соотношений и т. д. автоматизированы и не вызывают трудностей у неподготовленного пользователя, система имеет приятный дизайн, удобную, не вызывающую утомления систему управления 6-ю клавишами, доставляет радость в работе.

Основной принцип: пользователь должен только ввести информацию в первый раз - все остальное выполняется простейшим выбором из меню альтернативных предложений.

Несмотря на то. что освоение системы неподготовленным пользователем может проходить на интуитивном уровне, к ней прилагается краткая инструкция, гарантирующая освоение системы в считаные часы.

Эта система - наилучший ответ на Ваши самые срочные потребности. Хотите немедленно ощутить мойный эффект от внедрения ЭВМ в Вашу организацию - воспользуйтесь этой системой.

Введенная Вами информация не пропадет и при пере-ходе в дальнейшем на другие, более мощные системы, в тон числе и работающие в локальных сетях. Вся введенная Вами информация хранится в формате .DBF, который во всем мире признан неофициальным стандартом.

III. КОМПЛЕКТ ПОСТАВКИ Система поставляется на одной дискете 5.25" (KS DOS. 360 K). На ТИТУЛЬНОМ экране системы проставляется название вашей организации.

IV. ИСХОДНАЯ ИНФОРМАЦИЯ

Поскольку система исполняется персонально "под заказчика", от Вас необходимо получить исходную информацию - заверенный руководителем СПИСОК НЕОБХОДИМЫХ

ВАН ПОЛЕЙ ИНФОРМАЦИИ И ИХ РАЗМЕР, например:

1. ФАМИЛИЙ - 16 знаков

2 имя - 12 знаков

3. ОТЧЕСТВО - 16 знаков

4. ДАТА РОЖДЕНИЯ- в знаков

5. ДОН. АДРЕС - 40 знаков

6. пол - 1 знак

55. дущиЫАННЯ - йО знаков _____

ПРОСИЛ Вас:

1. Не задавать размер полей более 40 знаков. Например, поле АДРЕС можно разделить на 3 поля -ПОЧТ. ИНДЕКС. АДРЕС. ТЕЛЕФОН.

г. Не задавать названия полей длиннее 16 знаков. Пользуйтесь сокращениям». Например: ДАТА

ОКОНЧ. ВУЗА.

V. СРОК ИСПОЛНЕНИЯ ЗАКАЗА.

СРОК исполнения - г ~ 3 недели после поступления средств на наш р/с и заказа с заверенным списком ПОЛЕЙ ИНФОРМАЦИИ.

VI. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ К АППАРАТНО-ПРОГРАММНОМУ ОКРУЖЕНИЮ

1. Полная аппаратво-проГраниная совместимость с IBH PC XT/AT, надежность функционирования на отечественных модификациях не гарантируется и не обсуждается.

г. Наличие "жесткого" диска С•Винчестера") стан дартного объема.

3. ДИСКОВОД ГИбКИХ ДИСКОВ 5. 25" ИЛИ 3.5"". Эта
система поставляется нами без защиты от копирования.

4. Операционная система - MS DOS не ниже 3. 20.

5. Русификация компьютера в стандарте ГОСТ (кодировка альтернативная),

6. Требования к МОНИТОРУ - не специфицируются. желательно - EGA.

7. Требования к принтеру - совместимость со стандартом EPSOH.

VII. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

Гарантийным свидетельством при поставке программного продукта является картонный альбом, в КОТОРЫЙ вложены дискеты с указанной на нем датой продажи. при его отсутствии поставка выполняется с заверенным гарантийным талоном.

Гарантиями обеспечивается:

- бесплатная замена поставочных дисков, неработо способных в состоянии поставки (в течение месяиа после поставки);

- замена с минимальной оплатой при выгоде программ из строя по вине пользователя (механическое или электромагнитное повреждение, поражение ВИРУСОМ на машине пользователя и т. п.) или по истечении месяца после поставки. Нининальная оилата не превышает стоимость дисков + 5К текущей стоимости програмного обеспечения * стоимость почтово-транспортнык расходов и согласовывается с потребителем.

VIII. ПОРЯДОК ОФОРМЛЕНИИ ЗАКАЗА.

а) направить в наш адрес письмо-заказ с указанием необходимого программного продукта и количества ко пий. ПРИЛОЖИТЬ копию платежного поручения и заверенный СПИСОК ПОЛЕЙ ИНФОРМАЦИИ И ИХ РАЗМЕР.

Наш адрес: 10Т241, Москва. Б-241. а/я зт. "ИНФОРКОМ".

б) произвести предварительную оплату платежным поручением на наш р/с: Н 500461776 во фрунзенском коммерческом банке г. Москвы.

Стоимость системы на период март-апрель 1992г. - 7200 рублей +• 2вх

I "ЗЕЛЕНЫЙ ПАКЕТ" ДИСТРИБУТОРА

о тон, что такое "зеленый пакет" Вы можете

ПОДРОбИО ПРОЧеСТЬ В ЯП-12 "ZX-РЕВС" За 1991Г.

Стоимость "зеленого пакета" по системе "РЕГИСТРАТУРА" составляет для частных лиа: Рабочая версия программы - юи * 7200 - 720 Дискета - ioo Почтовые расходы - 5

Итого: 825 рублей

Уточняем, что высокая стоимость дискета вызвана тем, что нашим дистрибуторам поставляются дискеты особо высокого качества с защитным тег-~. новым покрытием, имеющие особый представительски вид и высокую коммерческую стоимость.

Напоминаем, что затраты дистрибутора на "золеный пакет" являются только залогом и возвращаются по требованию. АКТИВНО работающим дистрибуторам затраты возвращаются при выплате комиссионных и далее такие пакеты предоставляются бесплатно.

ZX - РЕВЮ
N3,N4 1992

МКП "ИНФОРКОМ" 121019, Москва, Г-19, а/я 16

Уважаемые читатели!

Кы обращаем Ваше внимание на изменение нашего почтового адреса. Все почтовые отправления просим направлять по адресу 121019, Москва, г-19, а/я 16. те. кто связан с нами не первый год, знают, что с этим адресом ми когда-то начинали и сейчас возвращаемся к нему опять.

Это временная мера. Ны подготовим новый постоянный почтовый адрес и в следующей выпуске вас оповестим.

I

в связи г этим мы вынуждены I прекратить прием от Вас предоп I латы за наши разработки. Все. что] Вы сочтете нужным заказать, будет! Вам высылаться наложенным плате I жом без предоплаты. I

Полностью также прекращается прием новых подписчиков. Если к концу года останутся нераспределенные экземпляры, предложим их I наложенным платежом. I I

Мы также приносим вам извине I ния за то. что Вы получите этот I номер со значительной задержкой, I вызванной сложностью с привлече- I нием доступных нам по затратам I полиграфических мощностей. Ны надеемся, что Вы простите нам эту задержку. вместе с тем, мы еще раз просим Вас не беспокоиться, все 12 выпусков в 1992 году Вы получите без каких-либо доплат, переподписок и т. п. Ны не исключаем возможности задержек, но все взятые обязательства выполним, невзирая на известные экономические трудности страна.

СПЕКТРУМ В ШКОЛЕ

К УРОКУ ИСТОРИИ.

Эту программу можно использовать на уроке истории для провср яи знаний. Ва. конечно, сани сообразите, как ее можно адаптиро вать для других учебных диспип лин. Правильный ответ выбирается путем нажатия на клавишу от 1 до ц.

Учитель может увеличить количество вопросов по своему желанию, добавляя строки DATA (500, 501, 50?... . . . 9999). Следует помнить, что в последней строке DATA должно стоять "eof, (end of file - конец Файла).

Ны использовали для диалога с пользователем ручсскни шрифт, полагая, что Ваш компьютер русифицирован. Если он не имеет русифицированного пзу, русифицируйте его программным путем, о чем на неоднократно писали, например в работе "Большие возможности Вашего "Спектрума"".

Программа будет также полезна начинающим для самостоятельного раэбора. Она достаточно проста для этой цели. Укажем только на небольшую особенность в строках

410 и 430.

Здесь Вам предлагается повторить ирру сгое раз. Если Вы хотите. Вы должны нажать клавишу "у. Поскольку неизвестно заранее, что I именно нажнет играющий "у" или "Ү". в строке 420 проверяется код символа, закрепленного за нажатой клавишей. Код "у равен 89, а код "у" равен 121. и в том и в другом случае выполняется переход к строке another (строка 110). после чего тест повторяется. Вы можете усложнить эту программу по своему вкусу, например рандонизировав (сделав случайным) порядок следования вопросов. Ноасете

ввести свою систему очков за правильный ответ с первой иопытки, за правильный ответ со второй попытки и штраф за неправильный ответ и т. п.

```
)0 8KH УРОК ИСТОРИИ. ЙО LKT data 480: LET finish 360: LET nextqiiest 130: LET another;1Ю
      30 REM оформление экрана 40 BORDER 2: PAPER 7: INK 9:
BRIGHT 1: CLS 50 PRINT PAPES 1; FLASH 1;
AT 9, 9 "УРОК ИСТОРИИ" 60 PAUSE 500 70 CLS
SO PRINT AT 4.2; "Эта программа служит для проверки Ваших знании по ИСТОРИИ. После
      каждого вопроса Вам предлагаются четыре возможных варианта ответа, только один из
      ко ТОРЫХ является зерним.
Выберите правильный ответ нажатием клавиши I...ч." 90 PRINT FLASH 1; AT 19,5; "Нажми
      любую клавишу" 100 PAUSE 0 110 REM another IEO RESTOffK data:
DIM Г(3>
130 REM nextiuest
140 CLS
150 READ a$
160 IF a$ "eof" THEN GO TO finish
170 KEAD bf.Ct, d*,e», f*
180 PRINT AT 6,2; 3* ' '
190 PRINT TAB 4; "1. ": b»
200 PRINT TAB 4; "2. -;c*
210 PRINT TAB 4; "3, ": d*
220 PRINT TAB 4; "4. ";e*
230 FOR n=l TO J
240 PRINT AT 2.2;
"Попытка ";n
250 PAUSE 0
260 IF INKEVt f*
THEN PRINT PAPER 1;
AT 19, 13; FLASH 1;
•ВЕРНО':
LET r(n)-r(n)»l:
PAUSE 150:
GO TO nextquest:
REM: правильный ответ
270 NEXT n
280 REM неверный ответ после
двух попыток
290 LET г(3)^г(3)*1
300 IF ?"-"!"
THEN PRINT AT 18,7; INK 2;
FLASH 1; b* II
310 IF f*- "2" ij
THEN PRINT AT 18,7; INK 2; II
FLASH l: c» Я
320 IF f* -3я Я
THEN PRINT AT 18.7; INK 2; Я
FLASH I: d* Я
320 IF f* "4" Я
THEN PRINT AT 18.7; INK 2; И
FLASH »; (?* 340 PAUSE 300 350 GO TO nextquesl 360 REM finish 370 CLS 360 PRINT AT 6,
      <?.;
"Правильных ответов t:
первой попытки "; г- (it 390 PKiNT AT 9,2;
"Правильных ответе» со
ВТОРОЙ попытки "; г (?) ЧОО PRINT AT lb, ?_, "Неправильных ответов "; г< 3) Н 410 INPUT
      "Попробуем еще раз"'". Я
у* П
420 IF CODK У* 89 \ OK CODK У* 121 II THEN GO TO another
430 .STOP
440 SKM ««»«»«"чч<"*«««««>«
• ДАННЫЕ » '.,.,.,.,.,.,.,.,.,.,.'. Ц
')50 REM II
• Введите столько дополнитель » ныл ВОПРОСОВ, сколько хоти
• те в строки 503 и далее. » за каждым ВОПРОСОМ должны » следовать четыре альтерна
• тивных ответа и цифра. п<жл
```

```
* зывасшзя кикой же из ник яв
* лнется правильным. Н « Список вопросов и ответов Н « должен заканчиваться строкОЙ N »
      DATA, n которой стоит запись В » "eof", как показано !) нашей
• примере.
500 DATA
"И каком ГОДУ и России было отменено крепостное право''",
-1825". "lew. "18Ы-. M91V.
-3"
501 DATA
"Кто из русских цлргй одержал победу в Полтавской битве?", "Иван Грозный". "Петр
      Первый". "Павел 1". "Николай 1" "2"
502 DATA
"Кто ип РУССКИХ царей был кавалером Мальтийского 11 Ордена". "Иван ГРОЗНЫЙ", "Петр |
      Первый". "Павел 1". "Николай 1"D
--Л^^Л*^^. _ I
```

POKES

NETHER EARTH

Наберите приведенный листинг и запустите его (RUN) перед загрузкой программы. Вы получите неограниченные ресурсы для создания боевых роботов.

Будет неплохо, если Вы перед запуском этого блока отгрузите его на ленту, чтобы избежать необходимости повторного набора, если досадная опечатка испортит Вам всю работу.

```
1 BORDER 0: PAPER 0: ПК T
5 CLEAR 65535
10 PRINT AT 10.3: "START 'NETHER
EARTH' TAPE*
20 LOAD "CODE 6*730
25 POKE 64753. 254
30 FOR f-65024 TO 65036
40 READ a: POKE f.a: NEXT f
50 DATA 62.18, 50. 111. 173
60 DATA 62.33. 50, 71, 174
70 DATA 195. 0,166
60 RANDOMIZE USR 64730
```

HEAD OVER HEELS

! Очень рекомендуем Вам сыграть в эту замечательную программу, нижеприведенный листинг позволит Ивам стать нпобедимым и нормально исследовать все ее лабиринты.

"ИНФОРКОМ" с удовольствием напечатает грамотно и художественно написанный отчет о вавен путешествии.

Программа снабжена счетчиком контрольной суммы и сможет вас предупредить, если при наборе вы где-либо ошибетесь, но копию все же лучше сделать.

```
1 CLEAR 64500
2 LET 1:0: LET w=l b FOR f:. 32000 TO 32170 10 READ a: POKE f. a 15 LET t^t*a»w: LET w =
      w*l PO NEXT f
25 IF 101764297 THEN PRINT "DATA ERROR": STOP
30 PRINT AT 10. l: "START 'HEAD OVER HEELS' TAPE"
50 RANDOMIZE USR 32000
100 DATA 221.33,203.92.17.234
110 DATA 6,62,255.55.205,86.5
120 DATA 48.241,243,237.94,33
130 DATA 44.125,229.33.173,98
140 DATA 289.51.51. 17. 163.252
150 DATA 1.22.3.33.253.94.62
160 DATA 202.237.79.195.173.98
170 DATA 33.70.125,229.33.199
180 DATA 252.229,51,51,17.209
190 DATA 252.1.г3г. 2.33. 209.252
```

```
200 DATA 62.196, 237.79. 195.199
210 DATA 252, 33. 209. 25г. 17, 309
220 DATA 138.1.92,0.237.176.33
230 DATA 228. 138. 34. 233. 13«. 3»
240 DATA 237.138.33.818,138.34
250 DATA 2*5, 138, 33, 255. 138. 34
260 DATA 9.139.62,195,50.29.139
270 DATA 33. I 16, 125, 34, 30. 139
260 DATA 195,209.136,175.50
290 DATA 166. 255. 62. 195, 50, 99
300 DATA 255. 33, 250, 250, 34, 100
310 DATA 255. 33. 145, 125. 17, 250
320 DATA 250, 1, 50. 0. 237. 176. 195
330 DATA 55.255,33.0.0.34,113
340 DATA 163. 33. 34. 25,34.115
350 DATA 163, 62. 33. 50, 120. 163
360 DATA 50.123.163,49,255,255
370 DATA 195,48. 112
```

INTO THE EAGLE'S NEST

Набрав этот блок Вы не только станете непобедимым, но еие и получите неограниченное количество боеприпасов и ключей от дверей замка.

запрос в строке 200 относится к неограниченному боекомплекту, можете при желании от него и отказаться.

строка 220 - выбор или отказ от непобедиимости.

Строка 240 - выбор или отказ от неограниченного количества ключей.

```
I CLEAR 25599
5 LET 1 = 0: LET W=0
10 FOR f=64000 TO 64037
20 READ a: POKE f, a
30 LET t = t*a>w: LET w=»»l
40 NBXT f
50 IF 1082517 THEM PRINT "ERROR IH DATA": STOP
100 DATA 33,14.250,17.0.91.1.50
110 DATA 0, 237, 176, 195. 0. 91
120 DATA 33.255,227,17.255.255. 1.0
130 DATA 128.237.184.175.58
140 DATA 32. 143. 58, 176. 160. 58
150 DATA 64, 158. 195,0. 128
190 POKE 23658.8
200 INPUT "IHFIHITE АИИО (V/H)? "; a$
210 IF a$="V THEM POKE 64026,50
220 IHPOT "INVINCIBLE IY/N)? "; a$
230 IF a$="Y" THEN POKE 64029.50
240 INPUT "INFINITE KEYS (Y/HI?"; a$
250 IF a$-"Y" THEN POKE 64032.50
300 PRINT AT 10.0: "START 'INTO THE EAGLE'S NEST' TAPE-
310 LOAD •" CODE
400 POKE 58380.26
410 POKE 58392,250
420 POKE 58695.100
450 RANDOMIZE USR 58368
```

URIDIUM

Эта программа позволит вам настроить игру на свой вкус.

Строка 110 - запрос на то. желаете ли Вы иметь бесконечное количество попыток.

120 - свободный пролет сквозь стены и т. п.

130 - игра без противников.

```
1 LET 1 = 0: LET w = 0 I 5 FOR f=64983 TO 65066
10 READ a: POKE f.a
11 LET t=t«a«w: LET W=V*1
12 NEXT f
15 IF t<>397017 THEN PRINT "ERROR IN DATA"
```

```
20 DATA 221,33.39.244.17
25 DATA 125.2.62.255.55
30 DATA 205.86.5.210.215
35 DATA 253,62.48.50,48
40 DATA 245, 33. 195, 0. 62
45 DATA 254,34.186.245.50
50 DATA 188,245,33,0.0
55 DATA 34,62,145,195.0
60 DATA 245.33.14.254.17
65 DATA 0. 64. 1. 40. 0, 237
TO DATA 176,195.0.64.33
75 DATA 255.239.17,255.255
80 DATA 1,0, 165.237, 184
85 DATA 62, 34, 58. 75, 138
90 DATA 62. 201, 56, 86. 152
95 DATA 62. 195,58.99,138
100 DATA 195.80.253 105 POKE 23658,8
110 INPUT "INFINITE HVES(Y/N) ?": a$ I
115 IF a$-"Y" THEN POKE 65051,50 I
120 INPUT "PASS OVER WALLS etc I CY/H)?"; a$ I
125 IF a$="Y" THEN POKE 65056,50
130 INPUT "NO ALIENS. LAND KOU PRINTED STRAIGHT AWAY 1Y/N)?": a$
135 IF a$="Y" THEN POГE 65061, 50
150 PRINT AT 10, .3:"START •URIDIUH' GAME TAPE-
200 RANDOMIZE USR 64983
```

AMAUROTE

```
1 CLEAR 26590
2 POKE 23658, 8
10 PRINT AT 10, 5; "START
'AHAUROTE' TAPE"
15 LOAD ""SCREEN»: LOAD "CODE
20 INPUT -INFINITE MONEY (Y/H)?";
a$
25 IF a$="Y" THEN POKE 46381.201
30 INPUT "INFINITE BOHUS (Y/H)?":
a»
35 IF a«*"Y" THEN POKE 40615. 0
40 INPUT "INVINCIBLE CY/H)?"; a$
45 IF a$--Y" THEN POKE 46312,0
50 RANDOMIZE USR 26600
```

GAUNTLET

```
1 CLEAR 64999
5 LET 1=0: LET w=0
10 FOR f=65000 TO 65032
15 READ a: POKE f.a
20 LET t=t*a«w: LET w=w*l
25 NEXT f
26 IF 1064702 THEN PRINT "ERROR
IH DATA": STOP
30 DATA 221.33.218.254. 17
40 DATA 81.1.62.255.55.205
50 DATA 86.5.48.241.33.1.254
60 DATA 34.57.255.243,195
70 DATA 0.355,62.201.50.82
75 DATA 184, 195.0. 132
80 PRINT AT 10.5:-START
'GAUNTLET' TAPE"
90 RANDOMIZE USR 65000
```

Продолжение. (Начало сн. на стр. 3).

Ознакомившись с общими чертами языка программирования БЕТА-БЕЙСИК з.о. ни теперь можем перейти к подробному рассмотрению его команд К функций.

Раздел 2. Команды

Команды приведены в алфавитном порядке

1. ALTER <атрибута> TO атрибуты

Ключевое слово расположено на клавише "A".

Команда ALTER позволяет выполнять значительные манипуляции с атрибутами экрана (INK, PAPER. BRIGHT и FLASH) для каждого знакоместа, в своей простейшей Форме команда ALTER может изменять атрибуты по всему экрану, не очи шая его,

Пример.

```
100 PRINT AT 10. 10; "TEST":
PAUSE 50: ALTER TO PAPER 1
```

эта строка изменит цвет PAPER всех символов на экране и сделает его синим. Можно провести по всему экрану установку и какой-либо комбинации атрибутов. ALTER TO PAPER 2. INK 7. FLASH 1

Нежно производить и выборочную смену атрибутов. Для этого надо перед то указать какие атрибуты Вы хотите поменять и на какие:

ALTER INK 7 то INK о

в этой примере все. что написано на экране белым светом INK изменится на черный.

Перед Вами открывается возможность создавать оригинальные видеоэффекты, например. Вы можете изобразить что-либо одинаковым цветом INK и PAPER и тогда изображение на экране будет не видно. Теперь командой ALTER вы меняете INK или PAPER и изображение проявляется перед Вами.

Попробуйте сделать вот такую головоломную комбинацию - она тоже будет работать: ALTER INK 3, BRIGHT I. PAPER 7 TO INK 5. FLASH 1

- но затронет только те знакоместа, в которых установлены INK-3, BRIGHT = 1 и PAPER =7 одновремен-но.

Следующая программа продемонстрирует некоторые из эффектов техники применения ALTER. пооро-буйте в ней поэкспериментровать с этим оператором. скорость мигания

полей на экране Вы можете изменять в строках 170 и гго, юо LET a=г, ъ^4

ПО FOR 1 = 1 TO 5

120 POK п=1 TO 16

130 PRINT IKK a: PAPER b;

"XXXX"; PAPER a; IKK b; -O000": 140 NEXT n 150 LET C=a. a=Ь. Ь=c 160 NEXT 1 170 LET t=30 ISO ALTER IHX a TO INK b

PAUSE t 190 ALTER PAPER a TO IKK a

PAUSE t 200 ALTER IHZ a TO PAPER b

PAUSE t 210 ALTER IHZ b TO PAPER a

PAUSE t

220 LET t-t - t/10 + 1 230 GO TO 180

2. ALTER <ссылка> TO ссылка

Ключевое слово расположено на клавише "A".

Это принципиально иная разновидность команды ALTER. По этой команде программа ишет появление первого значения и заменяет его на второе. Ссылкой здесь может быть имя переменной, число или строка символов. Например:

ALTER a$ TO b*

- заменит переменную a$ на Ь*; ALTER count TO c

- заменит переменную count на переменную c. но не затронет переменные accounts,

counter и т. п. ALTER I TO 23

- заменит число 1 на число 23. при этом, что очень важно, изменится также и 'невидимое' пяти-байтное представление числа, т. е. не только на экране вместо единицы будет изображено число 23, но и в расчетах тоже будет участвовать число 23. Однако:

ALTER 1 TO -23'

- заменит число 1 и его пятибайт ное представление на пару символов "23" и расчеты в программе будут производиться неправильно, хотя то, что Вы увидите на экране в измененных строках, будет записано вроде бы правильно.

Для всех вышеприведенных примеров по команде ALTER производится поиск по программе. при этом то, что стоит в кавычках. поиску не подлежит (предполагается, что Вы ишете и заменяете название процедуры, имя переменной или число). Но, в то же время, символьная строка будет найдена в программе где угодно, даже внутри кавычек, например: ALTER "breaK to stop" TO "any Key to stop"

Сами же кавычки при этом не разыскиваются. Если же Вам надо вместо одной из символьных строк использовать имя переменной, то

оно должно быть заключено в скобки для того, чтобы команда ALTER понимала, что Вы желаете изменить не имя переменной, а ее содержимое, например:

LET s* - "execute": ALTER is*) TO "execute"

Если же Вам надо использовать ALTER с символьными строками, содержащими ключевые слова, впечатайте их, воспользовавшись SVHBOL SHIFT/ENTER для того, чтобы принудительно включить курсор "K".

Вы можете использовать ALTER для быстрого удаления чего-либо из программы путем замены на пустую строку, например: ALTER "word" TO ""

Вы, наверное знаете (а в "ZX-РЕВЮ-91" мы этот вонрос обсуждэли). что числа требуют большого расхода памяти в БЕЙСИКЕ по сравнению с переменными и выражениями. Это происходит вследствие того, что после символьного вырахе-1 ния числа следует еще его пяти-! байтный аналог, и нередко в программах ПРОИЗВОДЯТ подмену, например вместо 1 употребляют liGN IM (экономятся 5 байтов) или. скажем вместо любого числа применяют VAL "число" (экономятся 3 байта). Это можно легко сделать с вомошьо ALTER. Напрмер, для чисел от 1 до 100.

```
1 LET free HEMO I
2 FOR П-1 TO 100 I
3 ALTER (П) TO -VAL" * CHR$ $4\ * STR* n * CHR$ 34 I
4 NEXT n I
5 PRINT "saved ": ЙЕНf) « 2ft I free: " bytes" l
```

Программа напечатает Вам гко I лько байтов ей удалось сэконо 1 нить. I

Переменная n в строке 3 стоит! в скобках для того, чтобы по ко 1 манде ALTER изменялось не имя no t ременной "п". а то число, которое! она хранит. I

Номера строк взяты минимальны I ни - от 1 до Ь и это сделано спе I циально. Дело в том. что при та t кой работе ALTER возможны сокра I пения расхода памяти в БЕЙСИК об I ласти и все строки могут "ное I хать" вниз в адресах памяти, а! это недопустимо для циклов FOR. . . I NEXT. Так что нельзя допускать.I чтобы до этого цикла в программе! могли бы быть какие-то строки.! способные измениться по ALTER. I

Число "28" в строке ь введено для того, чтобы компенсировать те! затраты памяти, которые потребо-1 вались на создание нужных в этой! процедур? переменных iree и п. т.е. чтобы эксперимент был чистым.

И последнее замечание. Будьте очень осторожны ори использовании команды ALTER в данной Форме. Неаккуратной работой Вы можете легко внести в программу неисправи-

мне изменения. Например, если вы изнешгге все переменные "apple* на "a", а потом сообразите, что "a" ухе использовалась в программе, то будет поздно. Вы не сможете сделать обратный код и снова поменять "a" на "apple", поскольку ври этой изменятся и те пере-нешше 'a", которые и деланы быть "a'. Можно прежде, чек делать такие замены, убедиться, что "a" в программе не использовалась -например командами 8EF или LIST REF (си. далее).

3. AUTO <номер строки> <,шаг>

Клавиша - "6".

AUTO - включает режим автоматической нумерации строк, что делает более удобным написание программ. ЕСЛИ значение шага не указано, предполагается, что шаг равен 10. Бели при этом не указан и номер строки, с которого начнется автонунераиия. то предполагается номер теклней строки (строки, в которой установлен программный курсор) плюс десять.

Режим AUTO отключается, когда номер строки менее ю или более 9963 или при выдаче любого системного сообщения. Обычно принято выходить из AUTO нажатием и удержанием BREAK продолжительностью более секунды.

Если, находясь в режиме AUTO, Вы хотите выпустить некоторый блок строк, сотрите предложенный программой номер строки и впишите вместо него свой собственный. То гда следующим номером, предложенным Вам. будет тот, что Вы ввели, плюс величина шага.

Примеры. AUTO - от текуией строки * to

с шагом через ю; AUTO 10О - от строки с номером

100 и с шагом через 10; AUTO 100,5 - от строки с номером

100 и с шагом через 5.

4. BREAK

Клавиши - CAPS SHIFT + SPACE в неграфическом режиме.

BREAK - не ключевое слово, Обчнзя команда BREAK из стандартного БЕЙСИКа вполне подходит для большинства приложений, но те. кто программируют в машинных кодах, знают, как часто программа входит в замкнутый цикл и не прерывается нажатием BREAK.

бета бейсик имеет резервную ввеак-систену. Если Вы нажмете и удержите SHIFT * SPACE более, чем в течение 1 секунды, эта система поймет, что вы "зависли" и исполнит BREAK даже если обычная система не работает. Этим методом можно выйти из затруднений, если вы неосторожно отключили 'STOP in INPUT" или "BREAK into program' посредством команды ОН ERROR.

Этот же прием выведет Вас из INPUT 1лнб. EDIT и AUTO.

Примечания:

1) Если Вы "вывалились' в исходное "Синклеровское" сообшевие. этот метод вам не поможет.

г) Программистам в машинных кодах, желатин воспользоваться работой резервной BREAK систенъ. не следует отключать прерывания. Система работает на прерываниях 2-го рода и они должны быть включены постоянно.

5. CLEAR число.

Эта команда должна представлять большой интерес для тех. кто работает с машинным кодом. CLEAR с числом, меньшим чем 767 перемешает указатель RAMTOP вниз на заданное этим числом количество байтов.

Почему 767? в этом числе всего три знака, поэтому Вы его никак не спутает с обычной командой CLEAR, после которой, как известно должен идти пятиразрядный адрес. Кроне того, это число легко реализуется на Ассемблере.

на экран, переменные и на стеки GO SUB, DO-LOOP и PROC эта команда не влияет. Клавиши определенные пользователем, а также область определения окон (расположенная непосредственно выше RAMTOP) перемешайтея вместе с RAMTOP. За этими определениями образуется свободное пространство, простираюпееся до начала кода бета-БЕЙСИКа 4707о. Обратите внимание: это пространство не заполняется нулями.

Та же команда CLEAR с отрицательным числом передвинет RAMTOP. и области определения клавиш пользователя и окон - вверх на заданной число байтов.

Никаких проверок на перекрытие машинного кода бета-БЕЙСИКа не производится, поэтому будьте осторожны и вообще не надо пользоваться этой командой, если вы ранее не перемешали RAMTOP вниз.

Маленький ньюанс. Когда Вы опускаете RAMTOP вниз на сколько-то байтов, образуется свободная область размером столько же байтов. Она образуется

непосредственно под машинным кодом БЕТА-БЕЙСИКа и вовсе не обязательно над RAMTOP, поскольку RAMTOP могла указывать в совсем иное место.

Пример позволяет вам поэкспериментировать с этой командой.

```
100 LET ram ЮP = 23730
110 PRINT DPEEK tranrtOP)
120 CLEAR 100
130 PRINT DPEEK (rarotoP)
140 CLEAR -50
150 PRINT DPEEK (raattOP)
```

6. CLOCK число или строка

Клавиша: "C-

Этот оператор позволяет управлять часами, показания которых могут быть показаны в правом верхнем углу экрана. Показания содержат часы, кинуты и секунды. Часы - в 24-часовом формате, может быть задействован режим таймера {будильника). В этом случае в заданное время может быть выдан звуковой сигнал, а может быть включена процедура GO SUB,

Часы работают от прерываний и потому счетчик работает и тогда, когда Вы пишете программу и тогда, когда Вы ее запускаете. Единственное, когда часы не работают - тогда, когда идет эагрузка/вы-грузка, исполняется BEEP или работает теневая периферия типа INTERFACE i,

параметром, определяющим в каком режиме работают часы, является число, стояшее после оператора. Это число может быть в диапазоне от о до 7 и имеет следуюаее содержание:

Число	ПереходGO SUB	Звуковой сигнал	Экран
0	НЕТ	выкл	выкл
1	НЕТ	ВЫЕЛ	вкл
г	НЕТ	вкл	ВЫКЛ
3	НЕТ	вкл	вкл
4	ДА	выкл	выкл
5	ДА	вопя	вкл
6	ДА	вкл	выкл
7	ДА	ВЕЛ	вкл

Часы начнут работать сразу после загрузки БЕТА-БЕИСИБа. На-чальная установка - "00:00:00".

выдать показание на экран можно командой CLOCK 1. Установка конечно не будет соответствовать истинному времени, но поправить дело можно командой

CLOCK строка , - например:

CLOCK '09:29:55'.

Здесь разряды отделены двоеточием, хотя в этом нет никакой необходимое ти. при вводе врене ни для установки, компьютер воспринимает только шесть цифр, а все символы-разделители (кроме символа "a") игнорирует. Если в Вашем вводе будут только пять пифр или меньше, он просто заменит недостающие нулем, например: CLOCK "xyzlO-

Здесь SYZ будут восприняты как символы-разделители и проигнорированы, а "10" - как установка первых двух разрядов. В результате получится: "10:00:00".

Символ "a", о котором мы упоминали, как об исключении, служит для установки таймера: "A06:2O" - установят будильник на Ь часов 20 минут. По достижении этого времени включится звуковой сигнал, если его режим был включен командой CLOCK г. з, б иди т.

Ножио сделать и так. что в данное время программа выполнит верекод GO SUB к наэаваченаой подпрограмме, если режим был включен CLOGS 4. 5, б или 7, Нравда такой переход возможен только в состоянии работающей программы. Если Вы в этот момент выполняете ввод или редактирование. Вашу работу прерывать компьютер не будет.

Подпрограмма, которая вызывается, может быть любой сложности и размера. Это может быть: ю GO TO 10. - а может быть текстовой редактор или игра. По достижении

заданного времени ком аьютер закончит ТУ строку, с которой он работает, а потом только сделает верекод. Завершение строки может занять определенное время, особенно если это INPUT или PAUSE.

Выбор и назначение подпрограммы, к которой выполняется переход GO SUB делается тоже оператором

CLOCK ЧИСЛО.

Злее ь "число" - это номер строки, к которой делается переход, это число должно быть и интервале от 8 до 9999. Нижний предел 8 назначен, чтобы отличать это назначение от выбора режима работы, в которой параметр может быть от 0 до 7.

Другой метод ввода подпрограммы для перехода по таймеру - прямой:

CLOCK: оператор: оператор:. .. : RETURN

В подпрограмме обработки перехода по таймеру нельзя использовать те же имена переменных. что и в главной программе, если вы не хотите, чтобы их содержимое изменялось. Вы можете оформить ЭТУ подпрограмму как процедуру и назначить используемые в ней параметры как LOCAL. Если Вы хотите, чтобы эта подпрограмма выполняла сохранение данный, а в главной программе возможно использование RUH или CLEAR, то Вам надо сохранять эти данные посредством POKE в соответствующих областях памяти, неоовреждаеиых по RUH или CLEAR.

Возможные применения подпрограммы перехода по таймеру могут быть такими:

- проигрывание музыкальной мелодии в заданное время (разновидность будильника);
- перестроение экрана;

бой часов (чтобы узнать при этом сколько сейчас времени, т. е. сколько раз должны пробить ваши часы, можно воспользоваться функцией TIKE* - см. далее.

вы можете даже изменять масштаб времени с помощью внутренней переменной бета-БЕНСИХа. размешенной в адресе 56866. Исходно там установлен режим хода часов 1/50 секунды на каждый ход.

роке эбвбб,56 сделает Вам режим 1оо секунд в минуте, а роке

56866.54 - вернет к нормальному режиму.

Специалисты в электронике могут организовать регулярный (ска-жем, каждый час или каждую МИНУТУ) прием данных от внешних устройств, например;

```
8999 STOP
9000 PRINT "Процедура включена' 9010 LET Polnter--DPEEK (USR "a"):
POKE pointer, 1Я 127 9020 LET pointer = pointer +1:
IF POinter>65535 THEN
LET Pointer - USR "a" * 2 9030 DPOEE USR "a".pointer:
LET 2* = TIKEK) 90*K> LET hours r VAL 2* (1 TO 2).
mins = VAL 2* (4 TO 5) 9050 LET mins = mins *!'
IF rains = oo THEM
LET hours-hours*1. mins - 0 9060 CLOCK "a" USIHG* ("OO",
hours) * USIHG» roO",roins):
RETURN
```

Не запускайте ЭТУ программу через йин. а вместо этого дайте прямую команду:

DPOKE USR -a" . USR "a" + 2: CLOCK 90001 CLOCK 5 эта прямая команда ПРОинициализирует указатель (pointer), расположенный в адресах USR "a$ и USR "a" м. строка 9000 назначена как строка перехода со таймеру. Команда CLOCK 5 делает этот переход возможным.

Для проверки задайте время срабатывания таймера:

CLOCK -AXXXX" - XXXX - установка времени.

Теперь запускайте некоторую программу через RUN. Подпрограмма CLOCK будет активироваться каждую минуту и считывать данные из вне-ннего порта 127. Прочитанные данные сохраняются в области гра+ики пользователя UDG в адресах, на которые указывает указатель (pointer). Указатель постоянно наращивается, а при достижении верхнего предела физической памяти 65535 возвращается в исходное положение. Если у Вас нет внешних устройств, подключенных к внешним портам, то можете организовать работу так. чтобы какое-нибудь другое полезное дело с помощью этой подпрограмм выполнялось через регулярные интервалы времени. строки 9040 - 90ЬО переуста-навдивают таймер на время,

отстоящее на 1 минуту от текущего времени, потом выполняется возврат в главную программу.

7. CLS <номер окна>

Просто команда CLS без указания параметра выполняет очистку
текущего окна. (Более подробно -
CM. WIHDOW).

CLS с параметром, выполняет очистку окна с номером, равным параметру (если конечно это окно было задано). Если же Вы забыли
определить это окно, то подучите сообщение об ошибке:
"Invalid I/O device"
(Неверно задано устройство ввода/вывода)

Команда CLS о - это то же самое, что и команда CLS стандартного БЕВСйКа. т. е. она выполняет очистку экрана вне зависимости от того, какое окно в настояний момент является активным.

в. CONTROL CODES IУПРАВЛЯЙ»!е коды)

Управляющие коды - это специальные символы, употребляемые в командах печати PRINT и PLOT, но сани по себе они не печатаются, а служат для того, чтобы указывать где что печатать. Так что в отличие от обычных символов их нельзя увидеть они не имеют графического изображения.

в бета-БЕИСНКе есть две группы управляющих кодов. Первая служит для управления курсором и познии ей печати по командам PRINT и PLOT, а вторая применяется лри управлении специальными блоками экрана, орабатываекыми по команде GET (см. ниже).

Коды управления курсором.

Коды	Наименование	Область действия
chr$ 2	курсор влево	экран
chr$ 3	курсор вправо	экран
chr$ 4	курсор вниз	экран
chr$ 5	курсор вверх	экран
chr$ 8	курсор влево	окно
chr$ 9	курсор вправо	окно
chr$ 10	курсор вниз	окно
chr$ 11	курсор вверх	окно
CHR$ 12	DELETE	окно
CHR$ 1ъ	Добавочный ENTER	окно

Разница между кодами с номерами 2. . . 5 и в. . . 1 1 в том, что коды, предназначенные для работы в окне, не могут вывести курсор «позицию печати) за пределы теку шего окна, это бывает очень удобным во многих случаях, например при создании текстового редактора.

Коды 2.. . 5 не имеют этого ограничения и применяются, как правило, с командой PLOT. Фактически же PLOT сама конвертирует коды 8. . . и в коды 2. , . 5 во время своей работы.

в стандартном бейсике обработка кода сне* s имеет ошибку. Так, невозможно перемещением курсора влево поднять его с нижележащих строк на вышележащие, а если он находится на самой верхней строке, то таким перемещением
можно его вообще вывести за предела экрана и зойти в распечатку программы. Здесь эта ошибка исправлена.

Код CHR$ 9 в стандартном БЕИСИКе тоже обрабатывается с ошибкой и здесь она тоже исправлена.

Стандартный бейсик распечатывает коды ю, 11. 12 в виде вопросительного знака, что не очень полезно. Здесь они работают так, как это должно бы быть.

& качестве примера рассмотрин. как управляющие коды, включенные в текстовые строки, позволяют выводить на экран сложные формы через PRINT или PLOT. ю LET a$ - "1234" * снк* в +

CHR$ 10 + -5' + CHR$ в *
CHR$ 10 * -678" * CHR$ в +
CHR$ 11 + "9"
го PRINT AT ю. ю; a$
30 PAUSE 100: CLS
40 FOR л= 32 TO 255
50 PLOT n, n/г: a$: BEXT n

Особенно полезным это может быть ери работе с графикой пользователя - эксперименты проведите сами.

CHR$ 15 работает, как ENTER в тон смысле, что позволяет прервать строку в любом месте и продолжить ее набор в новой экранной строке. Обычный ENTER послал бы строку при этом в листинг программы и работа бы с ней закончилась, кроме того. CHR$ 15 можно вставлять в текстовые строга, чтобы организовывать печать так, как вам это нравится. Вводится снк 15 одновременным нажатием CAPS SHIFT + ENTER.

Коды управления экранными блоками.

Этих кодов два: CHR$ о, CHR$ \.

Код CHR$ 0 говорит о том, что за ним еле джут восемь байтов. определяйте графический образ элемента экрана.

Код CHR$ 1 говорит о том. что за ним следует байт атрибутов и далее - восемь байтов, определяющие графический образ элемента экрана.

Эти управлявшие коды совместно с кодами снк* & и ю используются командой GET для сохранения блока экрана в качестве строковой переменной. Может быть Вам и не нужна нижеследующая информация, но кому-то она покажется интересной.

Когда команда PLOT встречает символ снк* о, она понимает, что очередные в байтов не являются печатными символами, а задают рисунок знакоместа размером 8X8 пикселей, который и должен быть воспроизведен на экране, это же относится и к команде PRINT, если задан CSIZE, отличный от нуля. (Аля нормального экрана следует задавать CSIZE равным восьми.) кодируется изображение знакоместа

по пмксельяын строкам абсолютно также, как это делается для символов графики пользователя, изображается состроенный шаблон в те-кушнх установленных аветах IHE и PAPER. Все. как с графикой пользователя ITOG. Более того. Вы можете создать массив из 9-тисин-вольнык строк, начинающихся с CHR$ 0 и использовать каждый элемент массива, как свой UDG-символ, понятно, что при этом Вы можете иметь огромные символьные наборы.

сня* I слегка отличается, т. к. за этим управляющим кодом идет еще один байт, определяющий атрибуты. поэтому при печати такого шаблона принимаются не текущие цветовые установки, а то. что содержится в этой байте.

Команда GET во время своей работы снимает изображение заданной области экрана, режет его на знакоместа, каждое знакоместо кодирует 8-ью или 9-ыо байтами, добавляет перед каждой серией байтов снк» о или CHR$ 1, вставляет между сериями символы управления курсором и. тем самым. представляет область экрана в виде длинной строки символов, которую и запоминает в какой-то переменной.

Если хотите поэкспериментировать с управляющими кодами, создайте строку и напечатайте ее сразу всю. целиком, а не по одному символу. 10 CSIZE 8 20 LET a$ = CHR» 0 + CHR$ 355 +
CHR$ 129 + CHR$ 129 + CHR$ 129
+ CHR$ 129 +CHR$ 129 +
CHR$ 129 * CHR$ 255 30 PRINT a$: P8INT CSIZE 16: a$:
PLOT 128.ев; a$

9. COPY строка
COPY массив

команда имеет очень близкое отношение к команде JOIH. Подробности см. в команде JOIH.

ю. CSIZE ширина <. выеста>
Клавиша: SHIFT + в.

CSIZE управляет размером символов при использовании операторов PLOT. PRINT и LIST. Аналогично 1нж и PAPER эта команда имеет глобальный характер, когда ис-пользуется в качестве самостоятельного оператора и распространяется только на один оператор, если стоит в качестве элемента в списке PRINT.

•ирнна и высота задаются в единицах пикселей. Если Вы не задаете параметр вые оты, то по умолчанию высота принимается равной ширине, нижеприведенный пример показывает символы, имеющие размеры в четыре раза больше, чем стандартные (CSIZE 32), но-. Вы можете сделать и символы во весь экран - CSIZE 255,176. Правда.

Для очень больших символов придется нажимать BREAK, чтобы остановить автоматический скроллинг экрана.

```
10 FOR n=8 то 32 STEP 8
20 CSIZE n
30 CLS
40 PRI1T "CSIZE "; n
50 LIST
60 NEXT n
70 CSIZE 0
```

Символы больших размеров выполняются пропорциональным увеличением стандартных "Спектрумовс-ких" символов. CSIZE 16 - увеличение в 2 раза, CSIZE E4 - в три раза и т. д. небольшие изменения в CSIZE могут повлиять на расстояние между символами при печати, не влияя на размеры самих символов.

Попробуйте в приведенном примере в строке 10 удалить оператор STEP 8 и посмотрите, что получится.

CSIZE с параметрами в,9.ю и 11 работает с шрифтом 8X8. а CSIZE с параметрами 12 ... 19 использует уже шрифт 16X16 и т.д.

Для некоторых параметров возможно, что при печати поля очередного символа будут перекрывать (затирать) рядом лежащий символ. Неплохой идеей борьбы с этим является использование при печати режима OVER 1.

Для выиеприведенного примера можно получить интересные эффекты, если перед командой LIST дать пряные команды:

INVERSE I: CSIZE 9

или CSIZE 32.8

или CSIZE 8. 32

Итак, мы рассмотрели работу с крупными символами, но можно поработать и с мелкими. CSIZE 3, 8 позволит Вам иметь 85 символов в строке, но это не очень зрелищно, разве что если Вы работаете только с малыми буквами и имеете хороший монитор.

CSIZE 4,8 дает 64 символа в строке. CSIZE 6.8 и CSIZE 7,8 используют стандартные символы и экономят на пробелах между ними. Если Вам надо иметь 40 символов в строке, делайте так:

OVER I: CSIZE &.8

После этого, с помошыо команды WINDOW уменьшите размер экрана по ширине до 24о пикселей и у Вас будут точно 40 символов в строке.

Подпрограмма, выподяяювая печать, отличается большой ухищренностью. Она может печатать в любой точке экрана (с координацией позиции печати до пикселя), в любом размере, согласует печать с активным в данный момент окном. Поэтому она работает конечно медленнее, чем стандартная PRINT-процедура. Для возврата к стандартной процедуре Вы можете использовать специальную команду

CSIZE о. При этом в качестве активного окна выбирается UIKDOW о (весь экран). После загрузки

бета-БЕИСЮЕа исходно!* является установка CSIZE о.

Все прочие коду управления печатью, такие как TAB. AT. запятая PRINT. коду управления курсором - работают в соответствии с установленных значением CSIZE. напринер, для режима CSIZE 4. в вы сможете дать такую команду, как

TAB 63.

Внутри операторов PRINT и PLOT CSIZE используется для создания временных

эффектов. 10 PRINT CSIZE в.16: -двойная
высота"; CSIZE *; "нормальная
высота": CSIZE 4.&; "малая
высота" 20 PLOT CSIZE За, 16; 100,100; "AS"

При печати символов графики пользователя есть небольшие особенности. Если ширина задана меньше, чем 6 пикселей, печатается только правая часть символа. Вы. конечно, можете подготовить свой набор символов, учитывающий этот факт.

Символу блочной графики обрабатываются как обычные символы -растягиваются, сжимаются и пр.

Jcть ограничения на печать блоков экрана, снятых по GET. Этот блок, как мы говорили выше, представляет собой строковую переменную, определяющую конструкцию шаблона 8X8 и взаимосвязь между соседними шаблонами. Поэтому бри печати таких блоков желательио использовать задания CSIZE, кратные восьми, в крайнем случае - 4. т.е. это 4.6.16.24 и т. д.

При использовании параметра 4 могут исчезать отдельные пиксели. но иногда для регулярных рисунков результат получается неплохим.

10. DEFAULT переменная - значение

<. переменная - значение>... клавиша: SHIbT + 2.

DEFAULT в принципе работает. как и LET, но в отличие от него не изменяет значение переменной, если она уже существует.

```
ю LET a=ю
20 DEFAULT a=30
30 DEFAULT b=30
40 PRIKT a. b
```

Строка 40 напечатает вам, что а= 10. поскольку DEFAULT в строке 20 будет проигнорирован, а b-30. т. к. DEFAULT в строке 30 сработа-ет.

DEFAULT следует понимать как выражение "принять значение, если иное не задано".

Как и за LET. в бета-БЕЙСИКе за DEFAULT могут идти множественные определения, причем обрабатываются они независимо, одно за дружин.

1оо DEFAULT a#="abcdef", ix-123,
4=0

этот оператор можно использовать в программе где угодно, во наиболее широкое применение он имеет в определениях процедур для того, чтобы пользователь ног опускать те или иные параметры при вызове процедуры.

п. DEFAULT = устройство

эта разновидность оператора DEFAULT позволяет задавать по умолчанию устройство ввода/вывода, она рассчитана на работу с
интерфейсом-1.

команда позволяет использовать обычные команды SAVE/LOAD/MERGE/ VERIFY при работе с иимродраквом, локальной сетью, глобальной сетью через порт HS232. То есть, благодаря еИ. упрощается синтаксис команд в работе с этими видами устройств.

в состоянии поставки бета-бейсик имеет установку "t", то есть на магнитофон. Примеры прочих установок:

DEFAULT = ш никродрзйв 1
DEFAULT = HI Кикродрайв 1
DEFAULT - ю2 НИКРОДРайВ 2
DEFAULT = п5 Локальная сеть, стандня н5
DEFAULT = в порт SS 232. канал "в"
DEFAULT = t Магнитофон

в качестве примера покажем, какие команды могут быть использованы после того, как была проведена установка DEFAULT = ш. все они будут работать с никродрай-вом 1. *

```
SAVE "test"
SAVE 1; "test"
SAVE »"m";1;"test"
SAVE 10 TO 100; "test"
LOAD "test"
```

```
VERIFY "test"
MERGE "test"
EKASE "test"
CAT
```

При этом, чтобы использовать микродрайв 2, можно либо задать DEFAULT = ma. а можно и не. задавать, а использовать:

SAVE 2, "test"
LOAD 2. "test"
ERASE г. -test"
CAT 2

Вы можете вместо номера использовать переменную, например:

LET X--2: DEFAULT па

Но нельзя использовать для этой пели строкОВУЮ переменную.

Даже если в качестве устройства ввода/вывода назначен магнитофон, такие команды как SAVE 1; *abc" иди LOAD n. "prog" будут работать с никродрайвом, поскольку заданный параметр номера одио-звачво указывает на то. что речь не идет о магнитофоне.

Текущие параметр» установки устройства будут выгружены вместе с кодой бета-БКЯСИКа. если Вы захотите выгрузить свою программу и бета-бейсик вместе с ней.

Примечание: бета-БВшСЛЕ разрешает в командах, связанных с някродрайвом использовать запятую "," вместо точки с запятой •;".

12. DEF KEY односимвольная строка; строка

или

DEF кет односимвольная строка; оператор: оператор: …

Клавиша: 1 (та же, что и DEF FH). см. также LIST DEF KEY

бета-бейсик позволяет присвоить любой цифровой или буквенной клавише значение символьной строки или программных строк. При этом символы могут вводиться в компьютер, а могут оставаться в нижней части экрана (в области редактирования) до нажатия ЕНТЕй.

Последний случай реализуется следующим образом: надо сделать так, чтобы последним символом Вашей строки стояло двоеточие ":" или чтобы последним оператором дрогаммаой строки стоял оператор ":". Попробуйте:

DEF KEY -1-; 'HELLO:"

Теперь нажните совместно SYMBOL SHIFT и SPACE. Курсор изменится и станет мигающее звездочкой. если вы теперь нажмете клавишу "1", в нижней части экрана появится надпись HELLO. поскольку никакие дпругие клавиши не били переопределены. при их нажатии вы получите их нормальные значения.

DEF KEY "a": PRINT "Goodbye"

в этом примере часть прогрэм мной строки, следующая после DEF KEY "a", присваивается клавише "a" (или "a", что одно и то же). Эта строка не исполняется после ее набора. После хе того, как Вы нажнете SYHB. SHIFT + SPACE и затем нажмете "a", она будет введена и исполнена, поскольку она не заканчивается двоеточием.

DEF KEY "a"; "1О REM hello"

Такая строка после нажатия на клавишу "a" будет реально введена в листинг программы после того. как пройдет проверку на синтаксис.

Клавише можно сделать переприсвоение в любое время, старое значение при этом будет переписано. Если присвоить клавише пустую строку или если после задания клавиши нет ни одного оператора, клавиша не будет иметь определения. Оператор DEF KEY ERASE уничтожит все сделанные определения, в том числе и те. которые раэне-вюам выше RAMTOP и, тем самым, защищен» даже от кем. процедура SAVE в загрузчике бета БЕНСИКа

выгрузит все определения клавиш вместе с нажиннокодовой частью бета-КЕИСИКа (она производят выгрузку кода от йАИТОР до конца бета-БЕЙСИКа).

Чтобы прснотреть все определения клавиш, пользуйтесь командой LIST DEF KEY. Если желаете подредактировать определение, присвоенное клавише, то можете выполнить следующий прием. Наберите номер строки, затем нажмите номер желаемой клавиши и Вы

получите редактируемую строку, которую Si сможете оформить в оператор DEF KEY.

RAMTOP автоматически понижается, когда Вы задаете клавиши. Если Вы воспользуетесь обычным CLEAR для изменения RAMTOP, то вполне может быть, что Ваши определения клавиш пропадут. с другой стороны. бета-бейсик имеет другой вариант CLEAR (си. выше), с помощью которого можно без риска для оиределении клавиш выделять пространство для своего машинного кода выюе уровня RAMTOP.

13. DEF PROC имя процедуры <параметрх, REF параметр>...

или DEK PROC имя процедуры <DATA>

Клавиша: 1 (та же, что и, DEF FH). См. также PROC. END PROC, LOCAL. DEFAULT, LIST PROC. REF. ITEH.

Оператор DEF PROC открывает определение пропедуры. Он должен быть первым оператором в программной строке (разрешаются только ведущие пробелы или предшествующие управлявшие цветовые коды), имя процедуры должно обязательно начинаться с буквы, а заканчиваться может пробелом, двоеточием, REF, ENTER или DATA. Имя про-иелуры может быть записано почти любыми символами, за исключением пробела, но желательно, чтобы Вы использовали буквы, цифры, знак подчеркивания "..", и, пожалуй Исинэолы "*г и "*". буквы верхнего Ни нижнего регистров - эквивалент-11ны. Процедура может иметь то же имя, что и переменная и путаница не произойдет.

За именем процедуры может идти список параметров или ключевое слово DATA, параметры, заданные в определении процедуры, называются формальными параметрами. Это имена переменных, когда процедура будет вызвана, то все имеющиеся в ней переменные с именами, совпадающими с формальными параметрами, будут зажишены и только потом ни будут присвоены значения фактических параметров, присутству-

ющих в вызове. Когда же перед именем формального параметра стоит REF. все происходит почти так же. но в добавок по окончании работы процедуры значение формального параметра будет присвоено соответствуыяену фактическому параметру, в процедуре могут участвовать и массивы, во они обязательно должны быть оформлены, как ссылки, то есть перед именем массива в определении процедуры обязательно должно стоять REF.

Когда вместо списка параметров стоит ключевое слово DATA, никаких переорисвоений не делается, а список фактических параметров при вызове процедуры может быть принят с использованием оператора READ и функции ITEH,

14. DELETE <номер строки> то <номер строки>.

Клавиша 7 (та же, что и ERASE).

Команда удаляет все строки в указанном блоке программы. Беля номер начальной строки оо?шен, предполагается минимальный, отличный от нуля. Если опушен номер конечной строки удаления, принимается номер последней строки программы.

примеры.

DELETE TO 100 - удаляет все строки после нулевой и до строки 100 включительно; DELETE 100 то - удаляет строку

100 и все послуцуюшие; DELETE 100 то 100 - удаляет только строку 1оо; DELETE 0 то 0 - удаляет только

нулевую строку;

DELETE TO - удаляет всю прогрзнну. за исключением нулевой строки.

Последний пример отличается от HEW тем, что не вычищает программные переменные. Все строки, явно указанные в операторе, должны реально существовать, иначе Вы получите сообщение об ошибке: U "Но such]ще" (нет такой строки). DELETE можно включать в текст программы, но с некоторыми предосторожностями, когда вышележащие строки программы удаляются оператором DELETE, расположенным в подпрограмме, процедуре, в цикле FOR... NEXT или DO... LOOP, программа обычно прекращает нормальную работу, поскольку запомненный адрес возврата ухе не соответствует реальным новым адресам строк. Если же оператор DELETE применяется для того, чтобы удалить самого себя из программы, он должен быть последним оператором в строке.

Возможное применение DELETE в программе - это удаление строк DATA после того, как они уже были прочитан» для того, чтобы освободить память для программных переменных, поскольку числа в строках DATA очень сильно расходуют память - по крайней мере яо в байтов на каждое число. Программа может также удалять часть строк для того, чтобы йодзэгрузкой выполнить MERGE нового блока на освободившееся место.

15. DELETE имя массива <предели> или DELETE строка <цредеяы>

Клавиша 7 (та же, что и ERASE)

Кроме удаления программных строю DELETE может удалять массивы полностью или частично я символьные строки. JO LET a$ ^'123456789" 20 DELETE a$ 14 то 7) 30 PRINT a$: йен Prints ме389" 40 DELETE a$

50 PRINT at: йен переменная не найдена

Когда удаляется строковая переменная, то она перестает существовать полностью, з не просто становится пустой строкой. Команда работает одинаково для строк и для массивов, создайте массив для эксперимента и попробуйте: ю DIM a$ (ю.9) 20 FOR п= 1 то 10

```
30 LET a$<HJ=CHR$(64t-n) *¨SXX¨
40 NEXT n
50 FOR HM TO length U,¨a$¨>
60 PRINT a$ Jn)
70 NEXT n
```

в строке 50 использована функпия LEHGTH вместо числа ю потону, что количество элементов в массиве будет изменяться. Теперь добавьте дополнительные строки:

```
45 DELETE a$ (3)
45 DELETE a$ (3 тоз
45 DELETE a$ (то 4)
45 DELETE a$
```

Опять запустите программу командой RUN и посмотрите, какие элементы будут удаляться. в числовом массиве команда DELETE а(б) удалит шестой элемент, если массив одномерный или целую строку элементов, если массив двумерный.

DELETE а(3 то и) удалят "вырезку" из элементов одномерного массива или группу строк из двумерного. (в числовом двунсрном массиве количество рядов задается первым числом.)

Продолжение следует.

МИХАЙЛЕНКО В. С.

ЗАЩИТА программ

Сегодня мы продолжаем разговор о приемах завита программ от просмотра.
Начало статьи сн. в "ZX-PEND"
HI-2. 199е г.. стр. 9-16.

2. 3.4. Управляющие коды, используемые для защиты от просмотра.

Естественно, что того описания, которое приведено в разделе 2. 3.з. недостаточно для полноценного использования управляющих кодов. Здесь иы более подробно рассмотрим их применение для защиты от листинга, изучая программы, взломанные известными какке раки, а также просматривая наиболее удачные программные приемы Фирменных программ.

итак ио порядку.

Код нт EDIT не используется для целей защиты, это управляющий код, который создает программа ввода с клавиатуры при нажатии клавиш "CAPS SHIFT" и "1" для вызова строки в область редактирования. Эта область изменяется в соответствии с размерами строки.

нв BACKSPACE. Этот код. означающий в переводе с английского "Курсор назад" достаточно широко используется для скрытия некото-рык частей строки Бейсика, для накладывания частей друг на друга с целью дезинформации и для создания необычных эффектов, один из которых был нами рас смотрен в примерах (раздел 2.3.2).

Естественно, можно использовав этот код сразу для нескольких целей одновременно. Рассмотрим теоретические способы использования BACKSPACE для скрытия информэоим и исполнения накладок (с практическими приемами Вы будете ознакомлены в следующей статье).

Предположим, первой строкОЙ в Вашей программе идет строка, выполнявшая реальные действия, например, осуществлявшая запуск программы в машинных кодах. встроенной в Бейсик (о том как сделать такую программу читайте в главе 1>. эта программа осуиеств-ляет реальную загрузку и только она может правильно загрузить, разместить и запустить блок кодов с магнитной ленты.

Чтобы ввести пользователя в текст этой строки скрывают, а в следующих за ней строках ставят ловушки (естественно, эти строки не скрывают - пусть все смотрят).

Ловушки могут быть самые разные -от продолжения программы на Бейсике, создавшей видимость загруз-ки до запуска программы в кодах в таком месте, где либо коды умышленно путаются с целью зависание компьютера, либо с отправкой на команду АССЕНБЛЕРа JP o000, аналогичной KANOOHIZE USR 0. что обеспечивает перезапуск компьютерной системы). Таким образом. вся сложность подобной системы защиты сводится к тону, чтобы освоить создание невидимой строки.

Делается это следующим образом: пишется нормальная строка на Бейсике, после этого она за-яуляется. чтобы ее невозможно было вызвать для редактирования, а потом "сжимается" до нуля влево с помошью символов BACKSPASE.

Примечание: еще больше затруднить поиск точного адреса старта Вашей программы в кодах можно используя измененный вариант числа управляющим кодом 14, который описан ниже, а кроме этого рекомендуется искажать подлинный смысл операторов Бейсика с использованием методов, описанных в разделе "Новейшие достижения защиты".

Более подробно как это сделано показано на рис. 1:

Рис. 1

где ни - номер строки нн - длина строки ABCDE - номер ячейки памяти, с которой осуществляется запуск программы в кодах,

После Бейсик строки мы размешаем ровно столько символов BACKSPACE чтобы текст строки был скрыт из виду т. е. каждому символу строки соответствует символ BACKSPACE.

Сокрытие лишь части строки из виду является разновидностью данного метода и используется в аналогичных целях, в частности, в широко распространенной программе СОММАДВО (Фирма ELITE» - оно используется следующим образом:

10 CLEAR 40000

во LOAD "" CODE

30 RABDOHIZE USR (неверный. т. е. ложный шаг. с которого якобы начинается программа в кодах.)

Достаточно хитро спрятан действительный адрес, по которому происходит запуск программы в кодах. Он размешен в строке го после команды загрузки LOAD "" CODE и "Сделан невидимым" с помощь» символов BACKSPACE. Для тек. кто интересуется более подробно, как это сделано, рекомендую составить программу для непосредственного просмотра ячеек памяти Бейсик-программы (дампкнга памяти), разместив ее в конце программы:

```
9997 FOR 1=23755 то зо000
9998 PRINT реек I; " ":снк реек I; " •; I
9999 NEXT I
```

Запускается эта программа командой GOTO 9997, а в случае остановки ее из-за невозможности распечатать тот или иной символ или по другим причинам, необходимо набрать с клавиатуры NEXT 1 и дампинг продолжится.

Этот метод рекомендуется применять для просмотра содержимого любых программ, т. к. он позволяет получить истинную картину Бейсика.

коды N9, ню. ни - RIGHTSFACE. DOWNSPACE и UPSPACE для ззииты программ не используются. Они выделены в отдельную группу т. к. генерируются программой ввода с клавиатуры для передвижения курсора в заданном направлении: "CAPS SHIFT" +6 - -> RIGHTJSPACfc -CAPS SHIFT" *D - -> DOWNSPACE U "CAPS SHIFT' +7 - -> UPSPACE I

коды HIг, 13 - DELETE и ENTER для защиты программ не используются.

N14: - этот управляющий код предшествует числам в программах. Как известно, в "спектруке- при программировании на Бейсике, обычные числа являются наибольшими расточителями памяти. Это происходит потому, что фактически числа записываются в память дважды -первый раз записано число в тон виде, в каком оно печатается на экране а второй раз записано истинное значение числа, т. е. то. которое обрабатывается интерпретзторон. Свидетельством того, что началась запись числа для интерпретатора и является управляющий код 14.

Введем, например, строку:

ю PLOT ю.9

и посмотрим, каким образом она запишется в память. Выглядит она так, как показано на рис. 2:

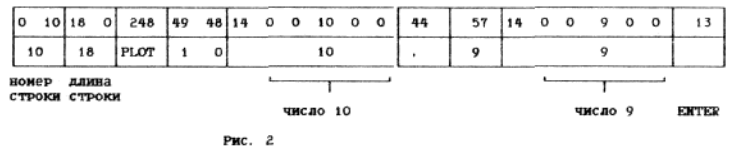


Рис. 2

Как видите, текст. который хранятся в памяти, отличается от того, что изображается на экране. После последней цифры каждого числа, выступавшего в тексте строки как параметр функции PLOT, интер-нретатор выделил & байтов ланяти и поместил таи символ с кодом 14. а за ним - 5 байтов: в которые записано значение этого числа (число записано способом, понятным интерпретатору, для * делающих более подробно изучить способы представления чисел в "спектруме" рекомендую обратиться к методической разработке "ИНФОРКОМа" "Первые шаги в машинных кодах).

это ускоряет в определенной мере выполнение программ на Бейсике, т, к. во время вшюлвения интерпретатор не должен каждый раз переводить числа из алфавитно-цифрового представления (просто последовательность цифр) в 5-ти байтовое представление, пригодное для вычислений на встроенном в ПЗУ калькуляторе "спектрума". Готовое значение выбирается из памяти после управлявшего кода 14.

Рассмотрим небольшой пример конкретной защиты программ. Во многих программах-загрузчиках присутствует такая строка:

о RANDOMIZE USR о: кем. . .

На первый взгляд эта строка после запуска должна перезапустить компьютер и. соответственно, очистить всю память "спектрума", но этого не происходит.

После более внимательного рассмотрения (с помощью реек-программы. которая была предложена ранее) оказывается, что поел е USR 0 и символа 14 нет 5 нулей -а именно так записалось бы число 0 в пятибайтяой форме. Эти значения были умышленно изменены, чтобы сбить вас с толку,

Но после управляющего символа 14 мы видим не нулевую комбинацию, а определенную последовательность, например: 0,0.218,92.0, что равнозначно числу 23770. т. е. практически функция RANDOMIZE USR Не осуществит переход по адресу о, а делает его именно на адрес 83770. а это как раз адрес байта, нэхо-Ндяшегося срззу после инструкции кем. где размешена программа в машинных кодах (о том, как разве 1).

н 15 - этот код "спёктрунои" в

стандартном БЕЙСИКе не исаольэу-ется. Использование его в других языках оговаривается в их описаниях, например в бета бейсик 3. о.

н 16 - код управления цветом символов - INK CTRL. После этого символа обязательно наличие одного байта, уточняющего о каком цвете

идет речь. Цифровая икала цветов совпадает со стандартной шкалой цветов спектруиа: с верный

1-сйний
2-красный
3-пурпурный
4-зеленый
5-голубой
6-желтый
7-белый

(в разделе "Секреты пзу" в прошлом году на стр. 14в "инфорком" указывал, что существуют также установки цвета с параметром 8 ("прозрачный") и 9 ("контрастный").

если после этого управляющего кода будет находиться байт, содержащий значение, не совпадавшее со значением одного из дветов, то это вызовет остановку компьютера с выдачей сообщения об ошибке INVALID COLOR (неверный цвет).

Создание системы команд, вызывающих остановку компьютера с выдачей сообщения об ошибке очень часто используется при защите Бейсик-ПРОграмм от просмотра. в частности, если мы используем этот метод совместно с методом зануления всех строк то это обеспечит полную защиту листинга, а кроме этого мы сможем оставлять свои сообщения в программах и закрывать доступ к ним (естественно, делать это стоит только на программах, созданных Вами лично) .

Рассмотрим более подробно организацию строки Бейсика при использовании управляющего кода !н? CONTROL. Как Вам уже вероятно известно, печать всех сообщений на экране "спектрума" осуществляет специальная процедура, встроенная в пзу. Когда эта процедура ветре -чает управляющий код 16. она анализирует следующий за ним байт и принимает его за числовое значение цвета INT, конечно если оно лежит в допустимых пределах.

После того, как код принят, все последующие символы на экране печатаются с цветом INK. соответствующим значению» стоящему после управляющего кода INK COSTROL.

в строках Бейсика применение управляющих кодов позволяет экономить память, что достаточно важно в программах и приобретает особую актуальность в загрузчиках.

Рассмотрим, откуда -берется эта экономия.

Традиционная форма записи Бейсик-строки: IMS 0.

Другой вариант - введение в строку управляющего кода INK CONTROL и за ним символа "о". Правда, набрать на клавиатуре эту комбинацию не так просто, о том, как это осуществить практически, смотрите ниже.

и в тон и в другой вариантах расходуется •приблизительно* одинаковое количество байтов памяти, за исключением того, что интерпретатор Бейсика переделывает число о в специальну» 5-ти байтную форму, с которой он и оперирует, то есть фактически при подаче команды INK о, расходуется не 2 байта памяти а а:

шк 0 14 0 о 0 о 0 , т. е. в 4 раза больше. Комбинация же:

INK COHTHOL о

расходует всего 2 байта. Поскольку здесь число о представлено в своем нормальном виде. Даже метод, использующий комбинацию:

INK нот PI.

'-что Фактически аналогично INK о) расходует 3 байта памяти, т. е. мы еще раз убеждаемся, что использование управляющих кодов является самыь экономичным из рассмотренных методов.

Еше более существенная экономия места в памяти достигается в том случае. когда управляющие коды используются в строке оператора Бейсика PRINT вместо вставного INK. Это объясняется тем. что в этом случае приходится учитывать символ ";" (точка с запятой)- которым необходимо "окаймлять" с двух сторон команду IHE.

10 FRIKT; INK о: "ZX-SPECTRUM" в данной случае под вставку команды IHE 0 расходуется 10 байтов:

; !к	о	14	0	и	о	0	0	
1 г	3	4	5	6	7	8	9	,0

в случае же употркбления управляющего кода INK CONTROL мы истратим лишь 2 байта.

Фактически это применение может использоваться не только для защиты программ, но и для элементарной экономии памяти, в частности, очень существенной эта экономия оказалась а программе "MOHOPOLY" (аналог известной во всем мире настольной игры], где управляющие коды используются для распечатки сообщений в операторе PRINT. Практически используя этот и другие приемы, программистам удалось создать великолепный экземпляр игры, в которую можно играть как в одиночку, так и в паре с соперником (под игрой в одиночку я понимаю поединок со "СПЕКТрУИом", когда компьютер выступает вместо второго игрока). Этот код дяет Вам возможность не только экономить память, но и создавать эффекты, достигнуть которые иными путями будет просто невозможно. в частности, хотите ли Вы. чтобы при загрузке вашей программы после ключевого слова PROGRAM текст названия вашей

программы печатается ишш (т. е. таким, который Вы заранее зада$ лиге) паевом? если да. то и об этом мы поговорим на страницах данных статей.

в 17 Управляющей код PAPER CONTROL используется точно так же. как и предыдущий код INK CONTROL, только в данном случае числовое значение следующее после этого кода изменяет цвет не символа, а •она.

в большинстве случаев управдя-кжие коды INK CONTROL и PAPER CONTROL используются вместе для достижения каких либо эффектов. Именно совместное их использование создает наибольшую выразительность и улучшает читаемость информации, а также позволяет достигнуть наилучших цветовых эффектов.

Рассмотрим более подробно теоретические аспекты совместного использования управлявших кодов INK CONTROL и PAPER CONTROL на базе примера 1. (си. раздел г. з. г>.

Вкратце навоняю, что тогда мы просто изучали эффекты, возникающие при использовании некоторых управляющих кодов без какой-либо предварительной теоретической проработки.

применение этих символов аналогично часто применяемым операторам изменения цвета INS и рарер на базе оператора FRINT.

Имеются, правда, существенные отличия.

Первое - это существенная экономия оперативной памяти (в некоторых случаях этот объем удается сократить в пять раз).

и второе отличие состоит в тон. что встроенный в PRINT оператор IHE или PAPER изменит цвет символов или фона лишь у текста. который должен распечатать PRINT. в то

время как соответствующие управляющие символы INK CONTROL и PAPER CONTROL изменяет цвет всего следующего за ними текста.

в частности, для эксперимента Вы можете вставить эти управляющие коды в один из фрагментов листинга программы способом. который будет описан в следующей статье (раздел J 3.5 - "Практическое применение управляющих кодов для защиты"}, в целях защиты это может применяться, например, в тех случаях, когда Вам нужно сделать текст программы невидимым) т. е. действие управляющих кодов IKE CONTROL и PAPER CONTROL аналогично действию постоянных операторов гне и PAPER.

Управляющий коп н18 задает или отменяет мигание - FLASH CONTROL. После этого управляющего символа следует байт параметра, кодировка аналогична стандартной кодировке. т. е. она может иметь значение о. 1

иди в. FLASH CONTROL I - ВЫЗЫВа-ет мерцание всех следующих после этого управляющего кода символов. FLASH CONTROL 8 оставляет в позициях символов прежние ранее установленные значения FLASH.

FLASH CONTROL 0 уничтожает действие предыдущих операторов FLASH

1 я FLASH 8. Поэтому ВС б индиюруеные после этого управляющего кода символы не мерцают.

Учтите. что FLASH CONTROL I заставляет мерцать позицию всего символа (8x8). даже в том случае, если высвечивается только одна точка.

Использование этого управляю-яего кода Вы могли наблюдать при загрузке програмны-копировшика "COPY-COPY" в тот момент, когда на экране появляется заголовок "PROGHAH", а после него следуют два слова программы, попеременно мерцающие.

Рассмотрим теоретические ас-пекты применения управляющего кода FLASH CONTROL. Его применение, как и других аналогичных управляющих символов, сопровождается значительной экономией памяти, в некоторых случаях за счет использования FLASH сонтвоС в сравнении с FLASH удается "выиграть" до в байтов памяти, что при достаточно частом использовании приносит ошутиную экономию.

Для того, чтобы заставить мерцать все символы. Вам необходимо лишь установить управляющий код перед первым из данных символов: FLASH CHTR 1 <текст програнмы>.

Если же Вы желаете. чтобы мерцал лишь небольшой отрезок текста (именно это необходимо в большинстве случаев), то Вам необходимо перед этим отрезком включить мерцание, а после него -выключить:

<текст прогрэммы> FLASH CHTR I <отрезок текста> FLASH CHTR 0 <текст программы>.

После ввода вышеприведенных кодов будет мерцать лишь <отрезок текста>.

Ничем не поддержанное использование управляющего кода FLASH CONTROL не принесет эффективности в защиту Вашей программы от несанкционированного доступа. Но применение этого кода может озадачить малоопытного 'каккера$. который столкнется с новым для себя приемом программирования.

Управляющий код N19 - BRIGHT CONTROL - индицирует следующие за ним символы более ярко. Это достаточно часто используется в игровых программах. но имеет очень слабую перспективу в применении для защиты программ.

После управляющего кода BRIGHT CONTROL следует байт параметра. который соответственно придает или отменяет более яркую окрэску в зависимости от своего значения:

BRIGHT CORTROL 1 дает более яркую окраску символам, выводимым после этого оператора.

BRIGHT снтн в указывает, что яркость символов в данной знакоместе должна остаться такой, какой она в нем была и ранее.

BRIGHT CHTR 0 устраняет действие BRIGHT 1 и BRIGHT 8. и далее символы индицируются с нормальной яркостью.

Теоретические аспекты применения BRIGHT CONTROL полностью энэлогичны использованию управляющего кода FLASH CONTROL.

Управляющий код н2о - INVERSE CONTROL меняет местами цвета в позиции символа,

т. е. цвет IHE становится цветом PAPER и наоборот.

После управляющего кода INVERSE CONTROL следует байт, определяющий применение данного управляющего символа:

INVERSE CONTROL 1 меняет местами цвета INK и PAPER у всех последующих символов, индицируемых на экране после этого кода.

INVERSE CONTROL 0. C00TBeTCT-венно. возвращает первоначальные установки.

Этот управляющий символ также как и BRIGHT CONTROL почти не применяется для защиты программ.

теоретические аспекты применения INVERSE CONTROL аналогичны применению FLASH CONTROL <сн. выше).

Управляющий код Nat OVER CONTROL используется тогда, когда необходимо индицировать символ, не уничтожая символа, который ухе находился в этой позиции, в зави синости от следующего после OVER CONTROL байта. т. е. того значения, которое находится тан. наполнение либо осуществляется, либо нет:

OVER CONTROL о. который д6йствует по умолчанию, уничтожает в индицируемой позиции ранее индицированный символ.

OVER CONTROL 1 индицирует символ, не уничтожая то. что находится в данной, и. таким образом, получается комбинация символов.

Использование этого управляющего кода может применяться для защиты листинга программ. в частности, совместное его использование с управляющими кодами BACK SPACE. AT CONTROL. TAB CONTROL, позволит осуществить наложение символов таким образом, что текст программы станет полностью нечитаемым.

Для примера рассмотрим теоретические аспекты совместного применения управляющих символов OVER CONTROL и BACKSPACE для создания готического щрифта. Если на алфавитные символы накладывать какой-либо простой символ. например

• /". (наклонная черта) > то мы ножен получить подобие готического шрифта, для того, чтобы осуществить наложение, нам необходимо напечатать сан символ, включить режим совмещения, сдвинуть курсор на одну позицию влево управляющим кодон BACKSPACE и после этого напечатать поверх этого символа наклонную черту. Последовательность действия:

в BACKSPACE OVER CHTR 1 /

Аля пояснения приведем алгоритм на Бейсике, аналогичный вышеприведенной строке. Введен строку:

PRINT "в"; CHR S:OVER 1; "/" после чего мы увидим, как наклонная черта перечеркивает "в".

Здесь был рассмотрен пример, позволяющий украсить шрифт, однако, нетрудно видеть, что можно умышленно накладывать некоторые символы один на другой так. чтобы исключить возможность прочтения листинга Вашей программы.

Код н22 (AT CONTROL) служит для печати сообщений в заданной знакоместе экрана. Компьютер по-Нлучает информацию о том. в каком месте осуществлять печать символов после анализа двух байтов, следующий: за рассматриваемым управляющим кодом. Первый байт но-

*ет быть в пределах от 0 до 81 и указывает, номер строки, в которй будут инфицироваться данные. Второй байт может лежать в пределах от о-до 31 и указывает номер колонки, начиная с которой будут индицироваться данные. Неправиль-1ное задание^данных значений вызо |вет останов компьютера по ошибке. I (под неправильным применением в 1данном случае подразумевается выход значений за допустимые пределы).

Управляющий код AT CONTROL является одним из наиболее употребительных и применяется для защиты не реже, чем INK CONTROL и PAPER COHT8OL. Возможности его применения очень широки и здесь все будет зависеть от вашей фантазии.

Рассмотрим теоретические аспекты на конкретных примерах, а к конкретному их применению вернемся нозже. Например. Вы хотите, чтобы после остановки вашей программы по команде BREAK и попытке листинга на экране появлялось заранее заданное Вами сообщение. Причем появляться оно будет в той позиции экрана, в которой Вы

желаете. Если хотите, то можете создать комбинацию, которая не будет выводить никаких сопроводительных сообщений - т. е. на экране не будет номера строки и других Бейсиковских атрибутов. Для достижения подобного эф-

фекта нам необходимо:

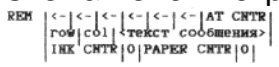
Во-первых, уничтожить Беясн-ковские атрибуты. используя управляющий код BACKSPACE.

Во-вторых, задать нес-то печати сообщения в необходимой нам месте экрана, используя управлявший код

AT CONTROL.

в-третьих изменить цвет INK и PAPER таким образом, чтобы дальнейший листинг программы не был бы виден.

Схема Бейсик-строки, позволявшей достигнуть этого:



Здесь:

<- - упр. код BACKSPACE row - номер строки, в которой должно печататься Ваше сообщение; col - номер колонки, с которой начнется ваше сообщение хаккеру, взламывающему вашу программу.

После того, как нам удастся создать эту строку практически, желаемый эффект будет достигнут. Подобное использование AT CHTR практически аналогично использованию AT совместно с PRINT. Но несмотря на это, применение управляющего кода позволяет получать эффекты, достижение которых иными методами было бы невозможно. Вышеприведенный пример является этому подтверждением.

Рассмотрим еще одно любопытное применение управляющих кодов, это касается использования их в заголовке программ, многие из Вас, вероятно наблюдали в некоторых программах появление названия программы (следующего после ключевого слова PROGRAM). высвечива-юшееся не в обычно принятом для этих целей месте, т. е. после слова PROGRAM, а в других местах, в частности, в центре экрана. Кроме этого, в некоторых случаях можно наблюдать появление заголовка вообше без слова PROGRAM (в этом случае название, как правило, печатается с начала строки). Все это достигается с помощью использования управляющего символа AT CONTROL.

Как Вам известно, хэдер (английское название заголовка, состоит из 17 байт, когда мы загружаем программу, то сначало идет короткий хэдер. а после него непосредственно загружается программа. Хэдер содержит информацию о типе программы* BASIC. CODE и т. д.). о длине программы, о месте размещения программы в памяти компьютера и другие подробности, различные для каждого типа загружающихся программ». Под название программы отводится 10 байтов. Этим объясняется тот факт, что мы не нолем записать на ленту программу с'количествен символов в названии большим, чей ю.

а теперь давайте представим ситуацию, когда мы вместо первых трех байтов в хэдере введен управлявши код AT COHTEOL (возможно введение и любого другого управляющего символа) с соответствующими значениями столбца и строки, а остальные семь байтов у нас будет занимать название программы. в результате этой операции мы получим возможность распечатывать название программы в любом месте экрана. Если же мы будем использовать другой управляющий код. то в зависимости от того, какими функциями он управляет, будем получать соответствующие изменения, недель данной строки: ;ат CHTR:row;coi: :::::::

1 2 3456789 10

первые три байта занимает управляющий символ AT CONTROL СО своими параметрами управления номером строки и столбца, а байты 4-ю отводятся под название текста программы.

Как видим, использование AT CONTROL несколько ограничивает наши возможности - в заголовке остается лишь семь свободных байтов для записи названия, но зато это создает неповторимый эффект. Кроме того, при удачном размещении в оставшихся 7 байтах хэдера ключевых слов Бейсика. Вам может удаться эффект создания полноценного названия, а в некоторых случаях общий объем печатаемого текста будет даже превышать 10 знакомест.

Код н23 (TAB CONTROL) используется для задания позиции данного, либо следующего столбца, начиная с которого в текшей строке печатаются указанные данные. После TAB CONTROL следует один байт, значение которого должно лежать в пределах от 0 до 31.

Основные аспекты использования TAB CONTROL для защиты аналогичны подобным моментам для AT CHTR с той разницей, что TAB CONTROL со своим параметром занимает всего 2 байта, в то время как AT CONTROL занимает з байта.

это обстоятельство определяет преимущественное отношение к TAB CONTROL в тех случаях, когда особенно важна экономия памяти, в частности, использование этого кода в хэдере дает выигрыш в 1 байт, правда несколько оскуднеет эФФект применения.

Ны рассмотрели использование всех, кроне одного, управляющих кодов. Не рассмотренный код COMMA CONTROL оставлен на саиыя конец повествования, ввиду того, что понять его работт лучше, если

знаешь уже теоретические основа применения AT свтв и TAB сету.

итак:

яод Нб - управлявший код COMMA CO8TROL (или управлявшая запятая оператора PRINT) используется для инднцйрования элементов, следующих после этого управлявшего кода на смешенных на пол экрана.

Более подробно алгоритм действия COMMA COBTROL можно понять нэ примере использования запятой в Бейсиковском операторе P8I8T.

Если в операторе PRINT элементы данных разделены запятыми, то они начинают индицироваться или с начала экрана иди с его середины.

Наиболее любопытным аспектом при использовании COMMA CONTROL является тот «акт что это не простое табулирование 16 символов, т. к. при этой очищаются все 16 знакомест, по который осуществлялась табуляция. Это значит, что COMMA CONTROL можно использовзть для стирания некоторых надписей на экране, особенно в тех случаях, когда необходима особая эко-вония памяти. Одним из этих случаев является применение COMMA CONTROL в хэдере. Именно с использованием этого управляющего символа удается уничтожить ключевое слово PROGRAM. которое должно появляться при загрузке Бейсик-блока. Достигается это с помошью полного "стирания" данного слова с поновью COMMA CONTROL.

в самом деле, если в 10 байтах яэдера, отведенных под заголовок, первой пойдет комбинация управляющих символов AT CONTROL выстав ляюшая позицию печати в начале ключевого слова PROGRAM. После этого это слово сотрется, с использованием COMMA CONTROL. Далее позиция печати переносится в любое удобное для нас место экрана, например, в его середину опять-таки с использованием AT CONTROL и ухе начиная с этого места осуществляется печать названия программы, схема распределения данных 10 байтов хэдера приведена на рисунке: AT сит»:row:coi:COMMA CHTR;

1 г з 4 AT CHTR!row;col: : : ;

5 6 7 в 9 10

Байты 1-3 занимает управляющий символ AT CONTROL со своими атрибутами.

Байт 4 занимает символ COMMA CONTROL осуществляющий табуляцию в середину строки.

Байт 5-7 занимает AT CONTROL с атрибутами.

и всего ливь 3 байта 8-ю остается под название программу.

Примечание " ИНФОРКОМа •: даже и в этих трех байтах ножяо разместить довольно длинные названия, используя токеиы ключевых слов.

Например:

ffETURH TO BUH

L AND HOVE (L * AND + HOVE)

DRAW CAT !

STEP OVER BORDER

танк (тая * BACKSPACE * к)

RUH TO POINT

GO TO HEW POINT
TABOR (TAB * BACKSPACE + OR)
OUTRUI
OVERLIST
и т. д. и т. п.

Несмотря на простоту, подобное использование управляющих кодов дос таточно и е экономично. но я надеюсь, что читатель в качестве эксперимента! укрепляющего свои познания в этой области, сумеет найти более рациональные решения, сохраняющие требуемый эффект.

2. 3. 5. Практическое использование управляющих кодов для защиты.

Ни рассмотрели теоретические основы применения управляющих кодов. я постарался дать исчерпывающую информацию обо всех, о тон. как это у меня получилось, судить Ван. читатель. в этом разделе перед нами стоит задача освоить предложенные теоретические идеи на практике. Если Вы хорошо поняли предыдущий материал, то и этот освоите без труда.

Итак, все по порядку.

Управляющие коды "СПЕКТРУМа" можно получить несколькими способами. Первый - самый легкий, понятный и доступный - использование встроенной функции CHR.

Как Вам уже вероятно известно, все существующие на клавиатуре ключевые слова и символы, а также определенные потребителем графические символы совместно с управляющими кодами образуют полный набор символов "ZX SPECTRUM".

Используя CHR и номер кода символа, можно получить строчное изображение каждого символа, такое правило будет соблюдаться для всех символов, кроме управляющих кодов. Если вместе с CHR мы наберем управляющий код. то никакой новый символ на экране не появится, а будет выполнено действие. соответсвуюшее данному управляющему коду.

Рассмотрим применение CHR с управляющими кодами на базе оператора PRINT. Существует несколько вариантов использования CHR. После CHR и значения указанного кода может следовать точка с запятой, например:

PRINT "A";CHR б;-в-

этот оператор выдает индикацию на экране:

а в

т. к. используется управляющий код COMMA CONTROL.

Использовать управляющие коды

CHS можно и иначе - выстраивая из них составную строку, так. оператор:

PRINT 'A' +CHR 6 +"в-

даст тот же эффект, что и приведенный ранее пример.

Как мы уже знаем. крон от 15 до аз управляют цветами и позицией печати. Каждый из них может использоваться в составной строке, после CHR 16 инк CONTROL} и

CHR 1т < PAPER CONTROL) должны

следовать CHR с указанным параметром цвета, а CHR IB. .. сне 21

(FLASH, BRIGHT. INVERSE и OVER CONTROL) должны применяться вместе с CHR о. CHR 1 или CHR 8.

Так. команда

PRINT CHR 16 « CHR 2 * CHR 17* CHR 6*CHR 18*CHR 1*"SINCLAIR-

изобразит на экране наднncь "SINCLAIR- мерцающую красным и желтым цветами.

Как и в случае, рассмотренном ранее, каждый знак плюс может быт заменен точкой с запятой.

После CHR Ей (AT CONTROL), как мы уже знаем, должны следовать два ключевых слова снк с параметрами, указывающими номера строки и столбца позиции печати.

Так команда

PRINT CHR гг* CHR и* CHR !6 *

"W"

отпечатает в середине экрана символ "V".

после CHR г з (TAB CONTROL > могут следовать 2 символа - это дает любопытный эффект: первое указывает позицию TAB. а второе обычно равно 0.

Так команда PRINT CHR 23*CHR 16+CHR 0+"S"

отпечатает в середине экрана символ "S".

как видите, такое применение данных кодов достаточно примитивно, а техника программирования, как Вы понимаете, не стоит на месте. Был создан более эффективный способ внедрения управляющих кодов, позволяющий обойтись без CHR. я не располагаю информацией о тон, кто и когда впервые начал использовать управляющие коды и кто впервые использовал их для защиты, но мне известно одно, а именно: ни одна современная программа к "ZX SPECTRUM" не обходится без применения управляющих кодов в своей защите, в связи с этим, возникают два вопроса:

1 Как научиться ставить защиту использующую управляющие коды, аналогичную применяемым в фирменных программах?

2 Как научиться быстро и просто снимать эту защиту с фирменных программ?

Рассмотрим теперь практическое применение управляющих кодов, итак по-порядку:

BACKSPACE

Незаменим, когда Вам необходимо скрыть какую-либо информацию Бейсика, например номер строки или некоторые другие атрибут», за

счет обратной табуляции Вы снохе-те также скрыть от посторонних глаз особо ценную информации.

Рассмотрим конкретный пример. Предположим у нас имеется строка вида:

1 Jен IT'S A FAHTASTIC TO USE BACKSPACE

а мы бы желали, чтобы во время листинга яа экране не было видно ни номера строки, ни Бейсик-оператора вея. Для этого нам необходимо изменить текст исходной строки, а именно: ввести между оператором REM и исходной надписью б символов (не имеет значение каких), чтобы потом заменить их на управлявший код BACKSPACE. Причем между вен и вставленными символами, а также между вставленными символами не должно быть пробелов.

Теперь нам необходимо символьную комбинацию, которую мы добавили между REM и текстом заменить управляющим кодом BACKSPACE. Для этсго вычислим адрес, по которому расположен первый символ данной комбинаций.

для этого используем известный нам факт, что область Бейсика на-

ловие верно только в тон случае, когда к компьютеру не подключен INTERFACE 1 с никродрайвом и некоторая1' другая переферия), в случае сомнений найдите этот адрес для своей системы сами, используя системную Беременную PROG (23635, г байта): рктнт реек г3635 * 256«реек 23636

Количество символов, которое нам необходимо пропустить, равно пяти (* символа это г байта номера строки и 2 байта длина строки • * код оператора REM).

Найдем адрес ячейки, в которой хранится первый символ нашей комбинации:

Ji3755*5=23760

Проверим это. подав команду: PRINT CHR реек 23760

после этого заменим зарезервированную комбинацию управляющий кодом BACKSPACE. Пододим команды с клавиатуры:

```
FOR I=23760 то е3765:
роке I,8:
NEXT I
```

После чего содержимое нашей команды исчезнет с экрана, но оно останется в памяти и будет обрабатываться интерпретатором.

для те?, кто желает оставить "надпись-памятку" для "хаккеров", рекомендую вернуться к примерам раздела 2. з. 2. (самый первый шаг).

Практическое применение управляющих кодов 1нж CONTROL и PAPER COBTROL лучше всего опис ывать вместе. поскольку именно такое их использование приносит наилучший эффект, итак, по-порядку. __

в защите применение INK CHTR и PAPER CHTR наиболее эффективно, когда нам

необходимо скрыть текст листинга. Достигается это с по-ношью подачи команд. задающих одинаковый цвет гж и PAPER. Рассмотрим более подробно, как это сделать.

Предположим. Вы желаете сделать так. чтобы после подачи команды LIST экран оставался пустым, для этого нам необходимо первой строкой программы поставить такую строку, которая бы "уничтожала" номер строки и изменяла цвета INK и PAPER управляющими кодами INK CONTROL и PAPER COHTKOL. чтобы стереть номер строки на экране нам понадобится использовать управлявший код BACKSPACE.

Итак, для начала зарезервируем место, используя оператор REM. чтобы потом спокойно заполнять его управляющими кодами, наберем

1 REM нннннннн 9символов

Нам необходимо зарезервировать место именно для 9 символов т. к. первые 5 заменит BACKSPACE, а остальные 2+2 мы используем для INK и PAPER COHTKOL с соответствуюшиии параметрами таким образом, чтобы сделать одинаковыми цвета символов и фона.

Для ввода первых пяти символов BACKSPACE подадим команду с клавиатуры:

```
FOR 1=23760 то 23764:
poke т, в:
NEXT I-
```

Для задания черного цвета иск подадим команды

```
poke 23765,16
poee 23766,0
```

Для задания черного цвета PAPER подадим команды

```
poke 23767,17
poke 23768,0
```

Теперь следующие вводимые после этого операторы не должны быть видны.

Введем первый оператор, создав новую Бейсик-строку и убедимся, что это соответствует действительности.

Теперь рассмотрим одно из наиболее любопытных применений управлявшего кода AT CONTROL. HHO-гин из Вас, наверняка доводилось встречать программы, при загрузке которых на экране появлялось лишь одно название программы (имеется ввиду, что отсутствовало слово PROGRAM), либо название программы появлялось в необычном месте экрана, например в середине. Достигалось это использованием кода AT CONTROL, а в первом случае с применением сонма CONTROL. итак.

по-порядку.

Для того, чтобы сделать такой заголовок, нам необходимо записать на ленту хэдер. содержащий управляющие коды.

поскольку, сначала необходимо создать такой хэдер, то возникает вопрос: "Как это сделать?", - ввиду того, что набор управлявших кодов с клавиатуры проблематичен. я предлагаю сделать следующее -после того, как Вы записали свою программу на ленту с любым названием, содержащий ю символов, загрузить ваш гэдер в копировщик copy - copy, найти там адреса ячеек, содержащих название, ис-пользуя команду LIST и поменять их командой poke, следуя нижеизложенным рекомендациям.

Для тех, кто не имеет копировшика COPY-COPY, возможен другой вариант. Одной из строк с воеА программы Вы делаете строку пп SAVE "имя программы" LIME в>.

После этого Вам необходимо получить дампннг памяти. используя одну из предложенный в этой главе программ, запомнив адреса ячеек памяти, в котрых находятся байты с названием Вашей программы, необходимо изменить их. следуя нижеизложенным рекомендациям, после чего обратиться к этой строке командой GOTO тш и записать файл на магнитофон, неудобство второго метода заключается в том. что созданная Вами строка nn SAVE "имя программы" LINE ю. должна будет остаться в программе, т.к. при записи яэдера в общую длину программы вошла также и длина этой строки.

Итак, какие надо делать изменения, чтобы название печаталось в произвольном месте экрана? необходимо лишь задать данную позицию, используя управляющий .код AT CONTROL. Из отводящиеся под название ю байтов, з будет занимать AT CONTROL со своими параметрами. Ввиду этого под само название остается лишь 7 байтов.

Дли того, чтобы распечатать название HEITFON в позициях AT ю,12, вам необходимо, чтобы в 10 байтах хздера. отведенных под название, содержалась следующая комбинация

гг	10	12	76	69	73	84	вг	79	78
AT с нтк	г	с	н	е	i	т	к	о	н
1	г	3	4	5	5	7	в	9	10

Если же ни хотим создать систему команд, уничтожающую слово PROGFAH. то необходимо действовать в следующей последовательности:

1) Установить курсор в позицию

AT 1,0 используя код AT CONTROL.

чтобы следующей командой стереть

это слово.

2) Стереть слово PKOGRAH. используя управляющий код COMMA COHTifOL.

3) Установить курсор в то место экрана, с которого кы желали ба распечатать текст названия программы с помощью AT CONTROL.

4) Распечатать название программы.

Таким образон, как видим. в предложенном варианте под текст названия программы остается всего 3 знакоместа. Для тея. кону этого окажется мало, можно не делать пункт 3 данного плана, что позволяет сэкономить ене 3 байта.

Этак, чтобы распечатать название программы HVS в позиции экрана в.е, уничтожив перед этим слово PROCRAM необходимо, чтобы 10 байтов заголовка имели вид:

« 1'	0	б	гг	в	8	77	86	s3
AT |г CHTR	с	соя. снтв	ATснтк	г	с	л	v	s
1 2	3	4	5	6	7	в	9	0

Возможно, что экспериментируя над управляющими кодами, читателю удастся обнаружить нечто новое в этой интересной области.

теперь некоторые советы, выше было ооисано использование управляющих кодов в операторе PRIKT через СвК. я полагаю, что Вам будет удобней экс п ериментировать именно исг->льзуя непосредственно управляющие коды.

и последнее, запомните, пожалуйста, что байт, стоящий после управлявшего кода, содержит именно значение данной цифры, а не ее ASCII код. Это правило касается абсолютно всех управляющих кодов.

nVOi'PAHHA для снятия защит

Предназначена для программ, которые используют управляющие коды 1ек COKTROL и гарев COKTROL.

```
9990 кен программист нихаяленко вадим йрти 010207, 1991
9991 PAPER 7;IHJ 0:BORDER 7:CLS y9y2 FOR 1=23758 TO 65000 9995 IF PEEK 1=13 THEN
IF PEEK(1+1)=39          AND
PEEK (1+2)=6 THEN
STOP
9994 IF PEEK 1-13 THEN LET 1=1+4
9995 IF PEEK 1=16 THEN
POKE (1*1),0: LET 1-1*2
9996 IF PEEK 1=17 THEN
POKE <I*I).7: LET 1=1*2
9997 NEXT I
```

Краткое описание.

Задав необходимые значения ПВетоВ ItS, PAPER и BORDER, программа а цикле начинает анализировать содержимое каждой ячейки памяти и. если встречается код IHX CONTROL то пвет символов принудительно задается черным, а для PAPER CONTROL - белым.

Строка 9993 следит за тем. чтобы как только программа дойдет до самой себя (до строки 99чх1}. произошла остановка. Поэтому наличие этой строки - обязательно.

Глава 3. Методы защиты от MERGE.

Среди приемов, наиболее часто применяемых для взлома программ, одним из распространеннейшкх является использование MERGE •вместо команды LOAD"". Это позволяет загрузить программу в память компьютера без ее автостарта.

Таким образом, в случае наличия в программе POKES, которые зашивают от BREAK, либо делают листинг программы нечитаемым, их удается разблокировать.

Автостарт должен был бы запустить программу, что позволило бы организовать зашкту посредством POKES, но MERGE загружает программу без автостарта.

Вот почему одним из первоочередных условий завиты является создание системы команд, способных защитить Вашу программу от использования MERGE".

Для того, чтобы уяснить себе, как действуют паяные методы защиты, необходимо проанализировать выполнение функции MERGE, итак, рассмотрим, как работает этот раздел интерпретатора Бейсика.

Как Вам уже вероятно известно, встроенные в ПЗУ программы обработки LOAD, SAVE. VERIFY и MERGE работают вкесте, используя многие обшие процедуры, команда MERGE очень близка по своей сути к команде LOAD, естественно с некото-рыки спеиифическими отличиями, в частности. HERGF. загружает файл Бейсика » специальный буфер, после загрузки интерпретатор начинает анализировать последовательно каждук строку загруженной программы. Если все проанализированное с точки зрения интерпретатора верно, то происходит непосредственное выполнение конайди MERGE, заключающееся в соединении двух программ в одну, причем новая программа затирает строки с тени же номерами, а также переменные с теми же именами.

Но это происходит лишь в тон случае, если до загрузки командой MERGE в памяти уже находилась какая-либо программа. Если же мы загружаем посредством MERGE новую программу в очищенную память, то. естественно, никакого слияния не происходит, а программа просто загружается в память, анализируется интерпретатором и после этого компьютер останавливает свою работу, поззоляя нам просмотреть листинг данной программы, так использование MERGE вместо LOAD позволяет загрузить программу без автостарта.

Быть может, теперь у нетерпеливого читателя возникнет желание просмотреть все фирменные программы, используя эту команду, что ж. попробуйте, но зеоб-хояино заметить, что в большинстве из них стоят защита от MERGE. Необжодиность такой защиты очевидна из-за описанных выше возможностей для хаккеров просматривать программы. Поэтому профес сионглы достаточно широко используют данный метод защиты.

Рассмотрим теоретические ас-секты работы защиты от MERGE,

Как вам уже известно, работа оператора HEXGE заключается в том. что он загружает программ? в специальный буфер и тан ее анализирует. Если анализ проходит успешно, то никаких сбоев не будет. но если он проходит не "очень гладко", то коньютер просто-нз-просто зависает. То есть, как видим, задача программиста, разрабатывающего защиту от MERGE, состоит в тон. чтобы его программа имела "встроенный дефект". который бы после загрузки с помощью MERGE проявлял себя, в то же время необходимо отметить, что этот "дефект" не должен никак проявляться после загрузки программы командой "LOAD".

Одним из методов, позволяющих осуществить подобную защиту. является задание фальшивого номера строки паралл ельно с Фал ыпивой длиной строки, фальшивый номер и длина строки сразу же проявятся. как только интерпретатор начнет анализировать программу во время выполнения команды MERGE.

Ко существует один нюанс, ко торый тоже необходимо учитывать. Интерпретатор анализирует и ту строку программы, которую он выполняет во время работы ирограи мы а это значит, что если после загрузки прогрлмиы интерпретатору Бейсика попадается созданная нами строка, то работа компьютера бу дет прервана по ошибке: "Ошибка в Бейсике {NOHSEN.4K IN BASIC)"

Следовательно, нам необходимо создать данную строку таким образом, чтобы она никогда не обрзба тывалась интерпретатором во время работы программы.

одним из методов, позволяющим сделать это. является размещение данной строки в

самом конце программы.

Рассмотрим, как это осуществить на практике.

Предположим, Вани создана программа- причем одним из необходи ных условий является то. чтобы нонера последних шагов программы не превышали 9960, поскольку в этой области памяти мы разместим специальную программу, которая создаст нан требуемую строку защиты от MERGE.

```
9965 FOR 1=23756 TO &5000 99fib IF PEEK 1=13 THEN
IF PEEK(I+1>=39          AND PEES (1*21=6 THEN POKE(I+1),255:
poee(1+гЬ255:роже(1+3».г55:
POKEII+4),255:
PRINT "THE E1D-: STOP 9987 NEXT I
9990 REM зажита от KERGE 9994 REM программист ннхайленко
вадни МЕНСК. яртя.
PP 010207.
```

Данная программа осуществляет принудительное изменение одного из номеров содержащихся в ней же строк, номер этой строки - 9990, так что наличие ее в программе - обязательно. Если читатель кочет оставить паиятку - сообщение для взломщиков, то он ножет написать в этой строке после REM произвольный текст, достаточно Убедительно показывающий нелосвешенно-му. что программа защищена именно Вами. Подобные комментарии постоянно оставляет после себя один из наиболее известных в нашей стране "хаккеров" BILL GILBERT, причем он использует именно этот метод защиты от KERGE"".

Программа действует следующим образом. в цикле анализируется содержимое текущей ячейки памяти. Если оно равно 13. то. следовательно, это код ENTER. а это, в свою очередь, не что иное, как окончание данной строки. а раз данная строка окончена, то. зная структуру Бейсик-программы. мы можем утверждать, что после этого кода следуют 4 байта, ответственные за номер следующей строки и за ее длину. Раз так. тогда анализируем нонер строки. Если его кодовое'представление равно 9990. - именно этот номер нам необходим, то тогда принудительно засылаем в ячейки Бейсика, ответственные за номер, значение е55. чтобы умышленно изменить содержащиеся там правильные значения, после того, как эта операция закончится, на экране печатается "THE END" и программа останавливается.

После того, как Вы создали "защитную" строку вы уже не увидите ее на экране. Но она будет последней строкОЙ и Вам необходимо строго соблюдать условие, чтобы интерпретатор Бейсика данную строку не анализировал.

После того, как предложенная программа выполнила возложенную на нее миссию - создала строку защиты от MERGE, ее необходимо уничтожить, подав команды:

9964 ENTER

9985 ENTER

9966 ENTER и т. д.

Заканчивая рассмотрение системы защиты от MERGE. хотелось бы подчеркнуть тот Факт, что здесь был описан лишь один из многочисленных приемов, применяющихся для создания защитной строки на Бейсике, препятствующих применению MERGE. Но то. что эта программа производит автоматически, можно осуществлять и •вручную". если использовать специальную програм-

му для получения "дампа" памяти. описанную в предыдущей главе.

Существуют методы защиты, достаточно кардинально отличающиеся от описанного выше.

ны уже говорили о том. что в бейсик-строке ножао разместить программу в машинных кодах, это используется с помощью применения оператора Бейсика кен и описано в главк I. Там же указано, что для нормальной работы интерпретатора необходимо, чтобы программа в машинных кодах ие содержала кода перевода строки - 13. это объясняется тем. что интерпретатор. анализируя данную строку Бейсика. определяет ее окончание именно по коду 13 и. если он его находит, то следующие за ним 4 байта он рассматривает, как нонер и длину следующей. бейсик-строки. а эти байты, естественно, не ногут правильно характеризовать бейсик-строку и. следовательно, должен произойти сбой в работе интерпретатора, это очень напоминает ситуацию, когда мы питались сделать фальшивыми номер и длину первой идушей в программе строки. То же самое происходит и в навей

программе, если мы используем в ней машинные коды, в которых есть код 13.

в этом есть некое рациональное зерно. Преимушества налицо, если у Вас есть программа, в которой на Бейсике сформирована строка машинных кодов, где присутствует код 13 (если он отсутствует, то его можно внедрить в программу, используя специальную директиву АССЕНБЛЕРа DEFB). в этом случае вам уже не понадобится Формировать специальную строку, ответственную за защиту от MERGE. что способствует экономии оперативной памяти, а это очень немаловажно для Бейсика, особенно если это загрузчик.

в то же время, следует учитывать, что этот метод иожет оказаться недействительным. если программа в кодах будет стоять в первой строке БЕЙСИКа. в этом и заключена одна из сложностей применения данного метода. поскольку интерпретатор начинает анализировать программу в нашивных кодах, как несколько строк бейсик-програнмы. разделенных кодом 13. в этом случае он может остановить свою работу и выдать код сообщения об ошибке:

NOHSEHSE IE BASIC,

Избавиться от проблемы можно, если сделать, чтобы машинные коды шли последней строкОЙ программы, осуществляется это достаточно просто: необходимо при создании своей программы лишь соблюсти определенную последовательность операций, которая и приведет к желаемому результату:

~ сначала Формируем строку с машинными кодами, но теперь при формировании даем ей такой номер, чтобы все строки налей исходной программы размешались до нее.

Причем необходимо, чтобы до начала следующего пункта нашего плана эти строки были сформированы. Здесь необходимо отметить тот факт, что даже в команде запуска программы в машинных кодах iKAHTOHIZE USR нннн; должны присутствовать значения цифр и именно такие, чтобы они приблизительно указывали на место расположения Вашей программы в кодах, это необходимо для того, чтобы Ваза программа имела точно фиксированное место расположения в оперативной памяти компьютера. после того, как мы в следующей пункте найдем точку старта подпрограммы в кодах, нам останется лишь заменить адрес в команде RANDOMIZE USR на правильный, но количество символов, соответствующих номеру старта не должно поменяться, т. к. уменьшение или увеличение числа цифр в номере приведет к смешению на один байт нашей программы в кодах. в обшей случае после команды RANDOMIZE USR должно сл6-довать пятизначное число, указывающее на какой-либо адрес, а после обнаружения правильного значения, это число соответственно откорректируется.

Для того, чтобы обнаружить истинное место в оперативной памяти нашей программы, можно использовать несколько методов:

1 - в частности, поручить это делать компьютеру, составив соответствующую программу.

2 - сделать просмотр памяти •вручную".

в любом из этих случаев нам понадобится вводить дополнительные строки в Бейсик, а делать это необходимо таким .образом, чтобы они размещались после строки, в которой содержится подпрограмма в кодах (используя такой прием, мы не изменим местоположение этой строки с подпрограммой в оперативной памяти).

в первом из рассмотренных случаев нам понадобится составить программу, чтобы она самостоятельно в цикле анализировала бы область Бейсика с тем. чтобы обнаружить там номер с троки в кодах. При обнаружение этой строки она должна распечатать номер с учетом 4-х байтов, отводящихся под номер и длину, и с учетом пятого байта, отведенного под REM. Выше было приведено достаточное количество подобных программ анализаторов, так что я надеюсь, что читатель сам справится с этой задачей.

Второй случай аналогичен первому с той разницей, что вам придется использовать программу получения даншшга. которая также была приведена в предыдущей главе.

Естественно, что после того. как эта программа выполнит свою задачу, ее строки необходимо будет уничтожить.

(Продолжение следует)

Дш . Х-эрдм^н, Э.Хьшзон перевод Тихоновой Н. А,

40 ЛУЧШИХ ПРОЦЕДУР

Ны продолжаем печать книгу Дж. Хардиаяа и Э. Хьюзова.

Сегодня Вашему вниманию предлагается подробный разбор еще 15 оолезных процедур для самообразования.

Начало см. •zx-РЕЕЮ" я1-г,
СТР. 17-28.

5.8 Сдвиг вниз на один символ.

ддияэ: 73

Количество переменных: о

Контрольная сумна: 79*7

назначение: Эта программа сдвигает содержимое дисплейного Файла вниз на 8 пикселей.

Вызов подпрограммы:

SAHDOHIZE us* адрес

Контроль ошибок: Нет

Комментарии: Нет

```
ЛИСТИНГ НАЯИННЫХ КОДОВ НЕГКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, 22527 33 255 87
            LD DE, 22495 17 223 87
SAVE        PUSH HL 229
            PUSH DE 213
            LD C,23 14 23
NEXT_L            LD B, 32 б 32
COPY.B            LD A, IDE) 26
            LD (HL).A 119
            LD A,C 121
            AND 7 230 7
                                  CP 1 254 t
            JR HZ,NEXT_B 32 2
            SUB A 151
            LD (UE).A 16
NEXT_B            DEC HL 43
            DEC DE 27
            DJNZ COPV_B 16 241
            DEC C 13
            JR Z, REST 40 El
            LD A, C 121
AH» 7 230 7

            CP O 254 O
            JR Z,K_BLOCX 40 24

            CP 7 254 7
            JR HZ.NEXT_L 32 225
            PUSH DE 213
            LD DE,1792 17 0 7
            AND A 167
            SBC HL,DE 237 82
            POP DE 209
            JR NEXTJL 24 215
REST        POP DE 209
            POP HL 225
            DEC D 21
            DEC H 37
            LD A,H 124
```

```
            CP 79 254 79
            RET Z 200
JS SAVE 24 201
K-BLOOE           PUSH HL 229
            LD HL,1792 33 0 7
            EX DE,HL 235
АНЭ A 167
            SBC HL,DE 237 82
            EX DE,HL 235
            POP HL 225
            JR NEXT_L 24 193
```

Как она работает:

В пару регистров HL загружается адрес последнего байта дисплейного Файла, а в OE загружается адрес байта, соответствующего изображению, отстоящему на восемь линия вверх. HL и DE сохраняются на стеке. В с-регистр загружается число, на 1 меньшее, чем число строк на экране. Затеи в в-регистр загружается количество байтов на одной линии дисплея - он используется, как счетчик.

В аккумулятор загружается байт с адресом DE и это значение пересылается в ячейку по адресу HL,

В аккумулятор загружается 1для проверки) содержимое регистра С и. если оно равно 1, 9 или 17. то в ячейку по адресу DE помешается 0. HL и DE уменьшаются на единицу, указывая на следуювий байт дисплея. Счетчик байтов в регистре 0 уменьшается и. если и он не равен О. происходит переход к СОРУ, а.

Далее уменьшается счетчик строк в регистре с. Если он равен о, происходит переход к процедуре 'REST'. Если С содержи! 8 или 16. то происходит переход к процедуре 'H_BLOCK'. Ксли С не содержит 7 или 15, подпрограмма переходит к 'NEXT-L'. Затем из HL вычитается 1792 - теперь HL указывает на следующую треть экрана и подпрограмма переходит к 'NEXT_L'.

В процедуре REST значения DE и HL берутся из стека и из ник вычитается число 256. в итоге DE u Hi. указывают на строку, позиция которой выше, чен va. что была в предыдущем шосле. Если HL содержит 20479, подпрограмма возвращается в BASIC, иначе происходит переход к процедуре SAVE.

в процедуре H_BLOCK 1792 вычитается из DE - т. е. после этого DE указывает на следующий блок экрана. Подпрограмма затем переходит к NEXT_L.

5.9 Сдвиг влево на один пиксел.

Длина: и

Количество переменных: о

Контрольная сунна: 1828

Назначение: Сдвиг содержимого дисплейного Файла на один пиксел влево.

Вызов программы:

RAKDOH1ZE USR адрес Контроль ошибок: нет

Комментарий:

Эта программа осуществляет более плавное перемещение. чем сдвиг влево на один символ, но требуется вызывать ее восемь раз, чтобы переместить дисплей на один полный символ.

ЛИСТИНГ НАЯИННЫХ КОДОВ

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,22527 33 255 87
            LD C,192 14 192
NEXT.L            LD B. 32 б 32
            ORA 183
NEXT^B RL (HL) 203 22
            DEC HL 43
            DJNZ NEXT-B 16 251
            DEC C 13
            JR HZ.NEXT..L 32 245
            RET 201
```

Как она работает:

В пару регистров HL загружается адрес последнего байта дисплейного файла, а в регистр С загружается количество линий в лис пдейион файле (он используется, как счетчик линий), в регистр "в" загружается количество байтов на одной линии (он используется, как счетчик байтов, флаг переноси устанавливается в о.

Байт, адресованный HL, сдвига ется на один бит влево, бит пере носа копируется в крайний правый бит, а крайний левый бит копируется во флаг переноса. Пара ре гистров HL уменьшается для указания на следующий байт, и счетчик (в-регистр) уменьшается, кс ли он не равен о. подпрограмма осуществляет переход к NEXT.. в. иначе уменьшается количество обрабатываемых линий, и, если оно не равно о. подпрограмма возвращается к процедуре NKXT_L. I

По окончании работы программа возвращается в BASIC. '

5.10 Сдвиг вправо на один пиксел.

Длина: 17

количество переменных: О

Контрольная сумма: 1Ъ50

назначение: Сдвиг содержимого

дисплейного файла на один пиксел вправо.

Вызов программы:

RANDOMIZE USR адрес

Контроль ошибок: нет Комментарий:

Эта программа осуществляет более плавное перемещение, чем сдвиг вправо на один символ, но требуется восемь раз вызывать эту программу, чтобы переместить дисплей на один полный символ.

Листинг машинных кодов

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, 16384 33 0 64
            LD C,192 it 192
NEXT_L            LD B, 32 o 38
            OR A 183
NEXT.B ЯК (HL) 203 30
            INC HL 35
            DJNZ NEXT.B 16 251
            DEC C 13
            JR HZ,NEXT_L 32 245
            RET J01
```

Как она работает:

i В пару регистров HI, загружа-1ется адрес дисплейного файла, а в С-регистр загружается количество линий на экране (он используется, как счетчик линий). В регистр "В" загружается количество байтов на одной линии (он используется, как счетчик сайтов}. Флаг переноса устанавливается в 0. Байт по адресу HL сдвигается на один оит вправо, бит переноса копируется в крайний левый бит, а крайний правый бит копируется во флаг переноса, пара регистров HL увеличивается, указывая на следующий байт, и счетчик (В-регистр) уменьшается. Если он не равен О, иодврограмма осуществляет переход к NEXT. В, иначе уменьшается количество обрабатываемых линий, и. если оно не равно О, подпрограмма возбраняется к NEXT L.

Закончив работу, процедура возвращается в BASIC.

5. 11 спвиг вверх на один пиксел.

Длина: 91

Количество переменных: 0 Контрольная сунна: 922в Назначение: Сдвиг содержимого дисплейного файла вверх на один пиксел. Вызов программы:

KAHDOHIZR USR адрес

Контроль ошиоок: Нет Комментарии: Нет

Листинг машинных кодов

```
НКТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
```

```
            LD HL.163в4 33 0 64
l.D DE. 16640 17 0 65
            LD C.192 14 192
НКХТ L 1.D B, 3?. б 32
СОРУ В            LD A, П>E> 26
Ml (HI.). A 119
I.D A, C 121
                        CP 2 J5* E
            JR H2,NEXT.B 32 2
            SUB A 151
            LD (DE1.A 16
NEXT_B            INC DE 19
            INC HL 35
            DJNZ COPY.B 16 J43
            PUSH CE 213
            LD DE, 224 17 224 0
            ADD HL, DE E5
            EX ISPJ.HL 227
            ADD HL, DE 25
            EX DE, HL 235
            POP HL 225
            DEC C 13
            LD A, C 121
            AND 7 230 T
                        CP 0 254 0
            JR HZ.       SUBTR 32 10
            PUSH DE 213
            LD DE,2016 17 224 7
            AND A 167
            SBC HL,DE 237 62
            POP DE J09
            JR H.BLCCK 24 14
            SUBTR
                        CP 1 254 1
            JR HZ,N_BLOCK 32 10
            PUSH HL 229
            EX DE, HL 235
            LD DE, 2016 17 224 7
            AND A 167
            SBC HL, DE 237 82
            EX DE, HL 235
            POP HL 225
N. BLOCK          LD A. C 121
АНЮ 63 230 63
                        CP 0 254 0
            JR HZ.       ADD 32 6
            LD A, T 62 7
            ADD A.H 132
            LD H. A 103
            JR NEXT_L 24 187
            ADD
                        CP 1 254 1
            JR HZ.NEXT^L 32 183
            LD A, 7 62 7
            ADD A.7 130
            LD D.A 87
            LD A, C 121
                                    CP 1 254 1
            JR HZ.NEXT_L 32 174
            RET 201
```

Как она работает:

В пару регистров HL загружается адрес дисплейного Файла, а в DE - адрес первого байта второй строки дисплея. В регистр 'C' загружается число линии на экране. В регистр "B" загружается количество байтов на одной строке -он используется как счетчик.

в аккумулятор загружается байт с адресом в DE и это значение передается в ячейку по адресу HL, в аккумулятор заносится содержимое регистра "с". Если оно равно ?., то DE указывает на нижнюю строку экрана и по этому адресу записывается о. HL и DE увеличиваются, чтобы указать на следующий байт, счетчик в регистре "в" уменьшается и. если он не равен о, происходит переход к copy_в.

224 добавляется к регистровым парам HL и DE, чтобы они указывают на следующую строку экрана. Затем уменьшается регистр с. счетчик линий. если содержимое регистра с не кратно в, происходит переход к SUBTR иначе 2016 вычитается из HL и происходит переход к H^BLOCK. Теперь KL указывает на следующие 8 линий.

в процедуре SUBTR- если значение ic-1) не кратно 3. происходит переход к H^BLOCK, иначе 2015 вы-читаеся из DE, т. е. теперь DE указывает на следующие 8 линий, в процедуре н.BLOCK, если значение с-регистра кратно 64. 1792 добавляется к HL и делается переход к NEXT.LIHE - теперь HL указывает на следующий блок из 64 линий, в процедуре ADD. если значение (с-1) не кратно 64. 1792 добавляется к DE, чтобы пара DE указывала на следующий блок 64 линий. Если с-регистр не содержит 1, то программа возвращается к H_LIHE. иначе - возврат в BASIC.

5. 12 Сдвиг вниз на один пиксел.

Длина: 90

Количество переменных: О

Контрольная сумна: 9862

Назначение: сдвиг содержимого дисплейного Файла вниз на один пиксел.

Вызов программы:

RANDOMIZE USR адрес

Контроль ошибок: Нет

Комментарий: Нет

Листинг машинных кодов

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,2252T 33 255 87
            LD DE, 22271 17 255 S6
            LD C.192 14 192
NEXT.L            LD B, 32 б 32
COPY_B            LD A, (DE) 26
            LD IHL1.A 119
            LD A.C 121
                                    CP 2 254 2
            JR HZ,NEXT.B 32 2
            SUB A 151
            LD IDE),A 18
NEXT.B            DEC DE 27
            DEC HL 43
.           DJNZ COPV_B 16 243
            PUSH DE 213
            LD DE,224 17 224 0
            AND A 167
            SBC HL,DE 237 82
            EX (SP).HL J27
            AND A 167
            SBC HL,DE 237 82
            EX DE,HL 235
            POP HL 225
            DEC C 13
            LD A,C 121
            AND 7 230 7
                                    CP 0 254 0
            JR HZ,       ADD 3J 8
            PUSH DE 213
            LD DE, J016 17 224 7
            ADD HL, DE 25
            POP DE 209
```

```
JB H-BLOCK 24 11
            ADD
            CP 1 E54 1 JH HZ.H.BLOCK 32 7
            PUSH HL 229
            LD HL,2016 33 324 7
            ADD HL, DE 25
            EX DE,HL J35
            POP HL гг5
H_BLOCK           LD Л,C 121
MID 63 . 230 63
                                    CP 0 J5'1 0
JH HZ.      SUBTR 32 Ь
            LD A,H -124
            SUB 7 214 7
            LD H,A 103
            JR NEXT_L 24 188
            SUBTR                                CP \ 254 1
            JR HZ.NEXT.L 32 184
            LD A, D 122
            SUB 7 J14 7
            LD D.A 87
            LD A,C 121
                                    CP 1 254 1
JH HZ,HJXT_L 32 575
            RET 201
```

Как она работает:

В пару регистров HL загружается адрес последнего байта дисплейного Файла, а в пару регистров DE загружается адрес байта линии, которая находится непосредственно над этим байтом. В регистр "С" загружается число линий экрана. В регистр "В" загружается количество байтов на одной строке дисплея - он используется, как счетчик.

В аккумулятор загружается'байт из ячейки с адресом в DE, и это значение загружается в ячейку по адресу в HL, В аккумулятор загружается содержимое регистра "С" если оно равно 2. то DE указывает на верхнюю строку экрана, в адрес которой заносится u. HL и DE уменьшаются, чтобы указать на следующие байты. Счетчик (Н-регистр) уменьшается и, если он не равен О, происходит переход к СОРУ „В.

224 вычитается из регистровых пар HL и DE - теперь они указывают на следующую строку экрана. Регистр С [счетчик строю уменьшается. Если содержимое регистра С кратно &, т. е. выполнено 8 ник лов для одной строки, происходит переход к процедуре ADD, иначе 20J& добавляется к HL и происходит переход к H..BLOCK - HL теперь указывает на следующие 8 линии.

В процедуре ADD, если значение (С-1) не делится без остатка на 8, происходит переход к ff BLOCK. в противном случае 2016 добавляется к DE, чтобы пара регистров DE указывала на следующие 8 линий. В процедуре И .BLOCK, если значение С-регистра делится без остатка на 64. из HL вычитается 1792 - HL теперь указывает на

следующий блок из 64 линии, и происходит перевод к NEXT_L. В процедуре SUBTR. если значение (С-и не кратно 64. из DE вычитается 1792 - в итоге DE указывает на следующий блок из 64 линий. Если С-регистр не содержит 1. то подпрограмма возвращается к NEXT-LIHE. иначе - в BASIC.

6. ДИСПЛЕЙНЫЕ ПРОГРАММЫ

6.1 Слияние картинок

Длина: 21

Количество переменных: 1 Контрольная сумма: 1709 Назначение: Эта программа объединяет картинку, хранящуюся в ОЗУ. с текущим экраном дисплея. Атрибуты не изменяются. Переменные:

имя - screen.store длина - 2 адрес - 23296

комментарий: содержит ^дрес картинки в ОЗУ. Вызов программы:

RANDOHIZE USR адрес

Контроль ошибок: Пет Комментарий:

Для объединения картинок должна быть использована приведенная на листинге программа. Однако ил тересные результаты могут быть также получены заменой OR (HL) на XOK !HL) или AND (HL).

Листинг машинных кодов

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,16384 33 0 Ь4
            LD DE,(23296) 237 91 0 У1
            LD BC,0144 1 0 24
NEXT..B           LD A, (DE) 26
OK (HI,) 182
            LD (HL).A 119
JNC HI. 35
            INC DE 19
            DEC BC 11
            LD A. B 120
            ORC 177
JK HZ. NEXT B 32 246
            RET HOI
```

Как она работает.

в ПЭРУ регистров HL загружается начальный адрес дисплейного файла, а в пару регистров DE -его длина. Регистровая пара ес используется в качестве счетчика.

в аккумулятор погружается байт с адресом в DE и выполняется логическое 'или' с OR') этого значения со байтом дисплейного Фай ла. Результат затем помешается в область дисплея.

HL и DE перемешаются на еле дуюрую позицию, счетчик уменьшается. Если счетчик не равен о. то подпрограмма возвращается назад для повторения процесса со следуюшнн сайтом.

6.2. Инвертирование экрана.

Длина: 18

Количество переменных: О

Контрольная сунна: 1613

назначение:

инвертирует все в дисплейном Файле: если пиксел включен - он сбрасывается (OP'K). если пиксел выгслючел. он устанавливается, Вы.гов программы:

RANDOMIZE USR адрес Контроль ошибок: Лет Комментарий:

Эта программа может быть использована для получения .эффекта вспышки Этот эффект усиливается, если делается вызов несколько раз и добавляется звук.

Листинг машинных кодов

```
НЕТ.*Л АССКMfc/Ш* ЧЧСЛА ДЛЯ ЬШЛА
! si ш,, 5 ь.т-t зз о м
i !* IX'-. b!4't 1 0 ?Л
1Л) П, 2!>5 22 L\b5 *
KEXT B            LD A, [) 1*2?
            SUB (HL) S'.iO
            LD ЦП.), A 119 II
            INC HI. 3b II
            DEC IJC 11 I
].D Л. B 120 I
ok i: 17/ • II
            JR N7,. ЯКХТ I) W. i'.4'f II          RET 201 I
```

Как она работает: I В пару регистров Hi. загружает I ся адрес дисплейного Файл;). а в! ВС загружается его длина И D ро I! ГИСТР помешается значение i>.bb, II Всякий раз. когда подпрограмма! возвращается к NEXT it. и ДККУМУ 1 ЛЯТОР загружается значение из II D регистра. Этот метол предпочти тельнее. чем ИНСТРУКЦИЯ 1.0 А-Н^, т. к. LD Л, D выполняется щ>и6лизи тельно в 2 раза быстрее, чем ИНСТРУКЦИЯ 1.D А. 2Ь5. Значение '>.jft та, хранящегося » ЙЧРИКГ по ЯДРО СУ, указанному в HL, пычитается из аккумулятора, а

результат за гружается в тот же сднии байт. Таким образом, делается инверти рование.

HL увеличивается, указывая на следующий байт, а счетчик, BC, уменьшается. Если счетчик не РД вен о, программа ноэшмшается к NEXT H. Если счетчик равен 0. программа возвратится и BASIC.

6.3 Инвертирование символа вертикально.

Длина: 20

Количество переменных: 1 I

Контрольная сунна: 17Ъ7 |

Назначение: Эта программа инвер S

тирует символ вертикально. I

Например, стрелка, направлен-I

ная вверх, должна стать стрелкой, направленной вниз, и наоборот. Переменные:

Имя - chr. start.

Длина - 2

Адрес - Ȣ296

Комментарий: адрес символа в

ОЗУ (ПЗУ). Вызов программы:

RANDOMIZE USR адрес Контроль ошибок: Нет Комментарий: Эта программа полез на в играх, т.к. можно изменять отдельные символы. не затрагивая при этом соседние области изображения

Листинг машинных кодов

```
ВЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,(23896) 42 0 91
            LD D.H 84
            LD E, L 93
            LD B. 8 08
NEXT_B            LD A, EHL) 12&
            INC HL 35
            PUSH AF 245
            DJNZ NEXT.B 16 г51
            LD B. 8 63
REPL        POP AF 241
            LD IDE).A 18
            INC DE 19
            DJNZ REPL 16 251
            RET 201
```

Как она работает:

в пару регистров HL загружается адрес данных символа. Этот іе адрес затем копируется в DE, В регистр В загружается значение 8 для использования регистра в качестве счетчика.

Для каждого байта в ЗККУНУЛЯ-тор загружается имеющееся в нас-тояший момент значение. HL увеличивается, указывая на следующий байт, а содержимое аккумулятора помешается на стек. Счетчик уменьшается, и, если он не равен о, подпрограмма возвращается, чтобы повторить процесс для следующего байта. В регистр "В" повторно загружаете я значение s, чтобы снова использовать его в качестве счетчика. изображение символа сохранено на стеке.

Процедура BEPL возвращает данные со стек& на то же знакоместоі ко уже з обратном порядке.

Байт за байтом берутся со стека и через аккумулятор помешаются по адресу, содержащемуся в DE, DE увеличивается, чтобы указать на следующий байт, а счетчик уменьшается. Если он не равен о, программа возвращается к KEPL. В ПРОТИВНОМ случае она возвращается в BASIC.

6.4 инвертирование символа горизонтально.

Длина: 19

Количество переменных: 1

контрольная сумма: 1621

Назначение:

Эта программа инвертирует символ горизонтально - например» стрелка, направленная влево, становится стрелкой. направленной вправо. Переменные:

имя - chr_start

Длина - 2

Адрес - 23296

Комментарий - адрес данных

символа. (гызов программы:

RAKDOHIZE U5R адрес КОНТРОЛЬ ошибок: нет комментарий: Нет

Листинг машинных кодов

```
НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
        LD HL,(2329G) 42 0 91          LD A. в 62 в ЫEXT_B            LD B, 8 68 NEXT.P
 EB (HL) 203 30 RL C 203 17            DJNZ NEXT_P 16- E50            LD (HL),C 113
            INC HL 35          DEC A 61          JR HZ, NEXT_B 32 243
        RET aoi
```

Как она работает:

В НЗРУ регистров HL загружается адрес данных символа, а в ак-КУНУЛЯТОР загружается количество байтов, которые должны быть инвертированы, в регистр В загружается число битов в каждом байте -он используется, как счетчик.

байт с адресом в HL сдвигается вправо таким образом, что крайний правый бит копируется во Флаг переноса, с-регистр сдвигается влево так, что Флаг переноса копируется в крайний правый бит. счетчик (В-регистр) уменьшается.

Если счетчик не равен О. происходит переход к NEXT_P для работы со следующим пикселом.

инвертированный байт, который находится в регистре с, помешается в ячейку, из которой он был взят.

HL увеличивается, указывай на" следуюпий байт, а аккумулятор уменьшается. Если аккумулятор не равен О, происходит переход к NEXT.BYTE. в противном случае -возврат в BASIC.

6.5 Вращение символа по часовой стрелке.

Длина: 42

Количество переменных: 1

Контрольная сумма: ЗвТб

Назначение: эта программа поворачивает символ на 90 градусов по часовой стрелке, например, стрелка, направленная вверх. становится направленной вправо переменные:

Имя - cbr_start Длина - 2 Адрес 23296

Комментарий: адрес данных символа. Вызов программы:

RANDOMIZE USR адрес

КОНТРОЛЬ ошибок: Нет Комментарий: Эта программа яолез-на в играх и для серьезных целей, например в II графике. II

Листинг машинных кодов

```
НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
        LD HL,(23296) 42 0 91
        LD E.128 30 128
K.BIT   PUSH HL 229
        LD C.O 140
        LD B, 1 61
NEXT..B       LD A,E IE3
AMD (HL) 166
                    CP 0 254 0
J8 Z.HOT_S 40 3
        LD A.C 121
        ADD A.B 128
        LD C,A 79
8OT_S SLA B E03 32
        INC HL 35          JR HC,NEXT^B 48 242
```

```
POF^HL 225
            PUSH BC 19T
SBL E 203 59
            JR HC.H_B1T 48 231
            LD DE, T 17 T 0
            ADD HL,DE 25
            LD B. 8 68
REPL        POP DE 309
            LD (HL).E 115
            DEC HL 43
            DJNZ REPL 16 251
            RET 201
```

как она работает:

Каждый символ состоит из ГРУП-чы пикселей размера в x 8, каждый из КОТОРЫХ может быть в состоянии ОН (1) идя OFF (О). Рассмотрим любой бит В2 байта В1 на Рис. 1. Данные хранятся в ячейке (В2.В1) в Форме:

j HI ИЗ I В2 Н4

где:

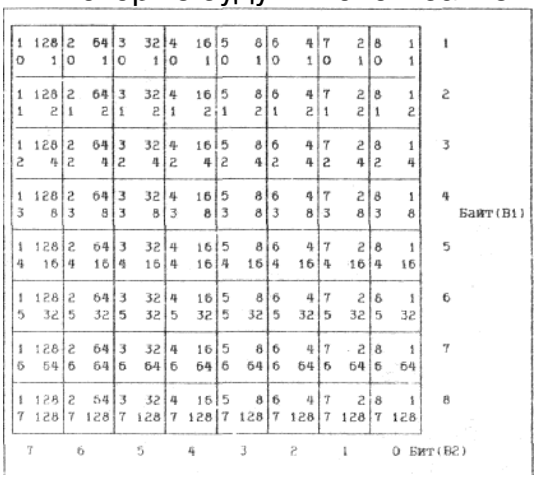
HI = байт, в КОТОРЫЙ пиксел (В2.В1) будет вставлен после вращения.

Н2 = бит в Hi, в КОТОРЫЙ он будет вставлен.

НЗ = значение, которое представляет текущее значение бита.

Н4 - значение бита Н2.

Каждый оайт врашаеного символа будет сформирован добавлением значении всех битов N2. которые будут в новом байте.



рис, 1 Ключ к подпрограмме вряшения символа.

В HL загружается адрес первого байта символа. В регистр Е загружается значение байта, КОТОРЫЙ имеет 7-й бит в состоянии ОН и с о по б бить - в OFF т. е. 128. HL сохраняется в стеке. В регистр С засылается 0. Далее в него будут добавляться данные, давая новое значение для формируемого байта. В регистр В загружается значение байта, нулевой бит которого включен, а биты 1-7 - выключены т.е. это единица.

в аккумулятор загружается со-дерхнмое Е-регистра (НЗ). Это значение перемножается логически (АНШ с байтом, адрес которого хранится в HL, ЕСЛИ результат равен О, происходит переход к КОТ-S. т. к. пиксел, адресованный регистром Е и регистровой парой HL сброшен (OFF). Если же он установлен (ОН), в аккумулятор загружается имевшееся значение байта (HI). Регистр В (К4) добавляется к аккумулятору, и это значение загружается в регистр С. Регистр затем устанавливается, чтобы указать на следующий бит HI. HL увеличивается, указывая на счедуюший байт <вп. Если байт Hi не завершен, подпрограмма возвращается к NEXT_B.

HL восстанавливается из стека, чтобы снова указать на первый байт символа. ВС сохраняется в стеке, чтобы запомнить значение последнего байта для завершения в С-регистре. Е~регистр настраивается на адрес следуююего бита каждого байта. Если врашение не завершено, происходит переход к N.BIT.

В DE загружается значение 7, и это значение добавляется к HI,. Теперь HL указывает на последний байт данных. В регистр В загружается число байтов, которые должны быть

взяты из стека, имя каждого байта новое значение копируется в E, и это значение помешается в ячейку с адресом в HL, HL уменьшается, чтобы указать на следующий байт и счетчик (B-Регистр) уменьшается. Если счетчик не равен O, происходит переход к REPL. Программа возвращается в BASIC.

6.6 Изменение атрибута.

Длина: 21

Количество переменных: 2

Контрольная сумма: 1952

назначение:

эта программа изменяет значение атрибутов веек символов экрана на задаваемое - например, могут быть изменены цвета всех символов, или весь экран может замигать и т. д. Переменные:

Имя - data saved

длина - 1

Адрес - 23296

Комментарий: неизменяемые биты атрибута.

Имя - new^data

Длина - 1

Адрес - 23E97

Комментарий: новые биты, вводимые в байт атрибута. Вызов программы:

KAHDOHIZE USK адрес КОНТРОЛЬ ошибок: нет

Комментарий: отдельные биты ЗТРИ-. бута каждого символа МОГУТ быть изменены с поиошью ИНСТРУКЦИЙ ASB и OR.

ЛИСТИНГ МАШИННЫХ КОДОВ

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, 22528 33 0 6в
            LD BC, 7бв 1 0 3
            LD DE, (23гУб) 237 91 0 91
NEXT.B            LD A,(HL) 126
AMD E 163
            ORD 178
            LD (HЬЬA 119
            INC H 35 i
            DEC BC 11 I
            LD A.B 120 !
            ORC 177 1
            JR HZ NEXT_B 32 246
            RET г01
```

Как она работает:

1 пару Регистров HI, загружается адрес области атрибутов, л в пару регистров BC количество символов на экране, в регистр D загружается значение new_data, а в РЕГИСТР E загружается значение data^sawed,

B аккумулятор загружается байт с адресом, находящимся а HI., а биты устанавливаются соответст венно значениям регистров D и к. Результат помешается обратно в ячейку с адресом, ХРННЯШИНСЯ в HL, KL увеличивается, указывая на следующий байт, а счетчик BC уменьшается. Если содержимое BC не равно 0, программа зозвраиает-ся к NEXT_B.

Программа возврашаетгя D BASIC

6.7 Смена атрибута.

длина: г?. \

Количество переменных: 2 [

Контрольная сунна: 1625 \

Назначение: Эта программа ишет [

атрибуты с определенным f

значением и заменяет каж-I

дое найденное вхождение
ДРУГИМ значением. 1
Переменные: I
Кия - old .value !
длина - 1 1
Адрес - 2329Ь
Комментарий: Значение байта, подлежащего замене имя - new_value длина - 1 Адрес - E3297 Комментарий: Значение замеша-
юяего байта, вызов программы:
RANDOMIZE USR адрес КОНТРОЛЬ ошибок: Нет комментарий: эта программа полезна для выделения областей текста и графических символов.
ЛИСТИНГ МАШИННЫХ КОДОВ

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, 22520 33 0 Вй
            LD BC, 768 103
            LD DE,(23396) 237 91 0 91
NEXT.B            LD Л, (HL) 126
                        CP E 187
            JR HZ.HO_CH 32 1
            LD (HU.D 114
HO_CH       INC HI, 35
            DEC BC 11
            LD A. B 180
            ORC 177
            JR HZ NEXT.B 32 245
HET 301
```

как она работает:
В пару регистров HL загружается адрес области, атрибутов, а в BC загружается число символов на экране, в E-регистр загружается old value, а в р-регистр -new_value. в аккумулятор загружается байт, адрес которого хранится в паре HL,. Если аккумулятор хранит значение, которое эквивалентно содержимому я-регистра, то в байт с адресом, имеющимся в HL, помешается содержимое D-регистра. при этом Ш, увеличивается, указывая на следующий байт, а. счетчик 1)C - уменьшается, Если BC не равно о, происходит переход к NEXT_B. иначе программа возвращается в BASIC.

6.8 Закрашивание контура.

Длина: 2&3
Количество неременных: 2 контрольная СУМНЛ: 26647 Назначение: Эта программа закрашивает область экрана, ограниченную линией пиксе леи. Иорешмшые:
Имя - x.-coord Длина 1 АДРЕС - 23296
Комментарии: Координата X
стартовой лозипии.
Имя V coord
Длина 1
Адрес 23297
Комментарии: координата V
стартовой позиции.
Визов программы:
RANDOMIZE USR адрес Контроль ошибок: ЕСЛИ коорди пата УМ75 или Ю1 NT (X, V) --I, то программа тотчас же возвращается в
BASIC.
Комментарий: Эта программа - HP
перонешаенля, ее стартовый
адрес - 31955. Когда зак
1 ративается большая область
I сложной ФОРМЫ, НУЖНО боль
I шое количество свободного

! пространства в ОЗУ. Если это невозможно, может произойти сбой.

ЛИСТИНГ МАШИННЫХ КОДОВ

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, (23296) 42 0 91
            LD A,H 124
                        CP 176 254 176
            RET HC 206
            CALL        SUBR 205 143*125
            AND (HL) 1&6
                        CP 0 254 0
            RET Яг 192
            LD BC,65535 1 255 255
            PUSH ПС 197
RIGHT LU HL,(23296) 42 0 91
            CALL        SUBR 205 143*125
            AND (HI,) 166
                        CP 0 354 0
            JR HZ, LEFT 3a 9
            LD HL, (23396) 42 0 91
            INC L 44
            LD (23296),HL 34 0 91
            JR HZ, RIGHT 32 236
LEFT        LD DE, 0 1700
            LD HL, (23296) 42 0 91
            DEC L 45
            LD (23296),HL. 34 0 91
PLOT        LD HI.. (Й3296) 42 0 91
            PUSH HI. 229
            CALL        SUBR 205 143*125
            OR(HL) 1S2
            LD (HL).A 119
            POP HL 225
            LD A,H 124
                                    CP 175 254 175
            JR Z, DOWN 40 44
            LD A,E 123
                        CP 0 254 0
            JR NZ.RESET 32 16
-           INC H 36
            CALL        SUBR 205 143*125
            AND (HI.) 166
                        CP 0 254 0
            JR NZ, RESET 32 7
            LD HL, (23296) 42 0 91
            INC H 36
            PUSH HL P.29
            LD E,1 30 1
RESET       LD HL,(23296) 42 0 91
            LD A,E 123
                                    CP 1 254 1
            JR NX,DOWN 32 15
            INC H 36
            CALL        SUBR 205 143"125
            AND (HL) 166
                                    CP 0 254 0
            JR Z, DOWN 40 6
            LD E,0 30 0
            JR DOWN 24 2
I, „JUMP          JR RIGHT 24 167
DOWH        LD HL, (23296) 42 0 91
            LD A,H 124
                                    CP 0 J54 0
            JR Z, NEXT.P 40 '10
            LD A, D 122
                                    CP 0 254 0
```

```
            JR Й2,REST 32 16
            DEC H 37
            CALL         SUBR 205 143M2S
            AND (HL) 166
                                       CP 0 254 0
            JR HZ, REST 32 7
            LD HL, (23296) 42 0 91
            DEC H 37
            PUSH HL 229
            LD D. J J21
REST        LD A,D 122
CF 1 254 1
            JR NZ,NEXT_P 32 14
            LD HL,(23296) 42 0 91
            DEC H 37
            CALL         SUBR 205 143*125
            AND (HL) 166
                                       CP 0 254 0
            JR Z,NEXT^P 40 2
            LD D, 0 ?.2 0
NEXT.P            LD HL,123296) 42 0 91
            LD A, I, 125
                                       CP 0 254 0
            JR Z,KETR 40 12
DF1C L 45         LD (23296),HL 34 0 91
            CALL         SUBR 205 143»125
            AND (HL) 166
                                       CP 0 254 0
            JR 7,PLOT 40 S29
            RETR         POP HL 225
            LD (23296), HL 34 0 91
            LD A,255 62 255
                                       CP H If18
            JR NZiL-JUHP 32 177
                                       CP 1 189
JE HZ. L,JUMP 32 174
            RET 201
            SUBK         PUSH BC 197
            PUSH DE 213
            LD A, 175 62 175
            SUB H 148
            LD H.A 103
            PUSH HL 229
            AND 7 J30 Y
ADP A, Ь4 198 Ь4
LO C.A 79
            LD A,H 124
R8A 203 31
                         RRA 203 31
                         RRA 203 31
            AND 31 230 3!
            LD B. A 71
            AND 24 230 24
            LD D,A 87
LU A.I! 124
            AND 192 230 192
            LD E. A 95
            LD H,C 97
            LD A,L 125
                         RRA 203 31
                         RRA 203 31 i
SRA 203 31
            AND 31 230 31 !
            LD L,A 111 1
            LD A,E 123 i
```

```
            ADD A. Б 123 I
            SUB D 146 I
            LD E.A 95 I
            LD D, 0 22 0
            PUSH HL 229
            PUSH DE 213
            POP HL 225
            ADD HL,HL 41
            ADD HI..HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            POP DE 209
            ADD HL,DE 25
            POP DE 209
            LD A,E 123
            AND 7 230 7
            LD B, A 71
            LD A,fl 62 6
            SUB B 144
            LD RA Tl
!,D A, 1 62 1
ROTATE            ADD A. A 135
            DJNZ ROTATE 16 253
SRA 203 31
            POP DE 209
            POP BC 193
            RET J01
```

Как она работает:

Эта программа вычерчивает горизонтальные линии из смежных пикселей. Назовем их строчками. Предел заполнения области строчками ограничен включенными (ОН! пикселями. Каждая строчка запоминается с помошью занесения в стек координат крайнего правого пикселя этой строчки.

Запускаясь с определенных координат, программа делает заполнение в каждой строчке, отмечая позиции каждой из невыполненных строчек выше или ниже, по завершении одной строчки последнее значение отмеченных координат восстанавливается и для соответ-ствуюшей строчки происходит заполнение. Этот процесс повторяется до тех ПОР. пока не останется незаполненных строчек.

Рис. г иллюстрирует технику процесса, квадраты представляет пиксели, X обозначает стартовую позицию в пределах области штриховки, а • обозначает крайние правые пиксели строчек.

Рис. 2 Иллюстрация техники заполнения области. X - это стартовая позиция, « - начало строчек 0 - оставшаяся незаштрихованная область.

ПРОграмма штрихует горизонтальную линию, содержащую стартовую позицию и сохраняет в стеке позицию начала строчки иа линиях непосредственно выше и ниже. Далее она штрихует линию выше, а затеи ниже, отмечая в последнем случае, что еще две строчки запускаются на следуюшей нижней строке и т. д. любая позиция в пределах области для штриховки может быть выбрана как старто-

вая позиция. Но заметим, что два пиксела, промаркированные нулями. нетронуты, т.

к. они отделены от заштриховываемой области.

В регистр Н загружается У-ко-ордината. а. в L-регистр - X-координата. Если значение Y больше, чем 175. программа возвращается в BASIC. Процедура SUBR вызывается, возвращая адрес бита (X.У> в память. Если этот бит в состоянии 'OH' (включен), подпрограмма возвращается В BASIC.

число 65535 помешается в стек, чтобы отметить первое сохраненное значение. Позднее, когда число восстанавливается из стека, оно интерпретируется, как пара координат. Однако, если число равно 65535, происходит возврат в BASIC, т. к. программа закончена.

В регистр Н загружается У-координиата. а в регистр L - X-координата. процедура

SUBR вызывается, возвращая в HL адрес бита (X.Y). Если этот бит установлен (OH), происходит переход к LEFT. Иначе X-координата увеличивается, и делается переход к RIGHT, если X не равен г5б.

В процедуре LEFT DE Устанавли вается в 0- Регистры си Е используются, как Флаги: D вниз (DOWN). Е - вверх (UP). X-коорлината уменьшается. SUBR вызывается, и вычерчивается точка (X.Y). Если Y-координата равна 175, подпрограмма переходит к DOWM. Если флаг "ВВЕРХ" Установлен. происходит переход к RESET. ЕСЛИ бит (X.Y+1) сброшен, значение х и У*1 сохраняются в стеке и флаг "ВВЕРХ" включается.

В процедуре RESET, если флаг "ВВЕР.Х" включен, происходит переход К, DOVH. ЕСЛИ бИТ (X.Y+1) включен (OM). Флаг "ВВЕРХ" выключается, в процедуре DOWK, если Y-K00P-дината равна о. происходит переход к NEXT_PIXEL. Если Флаг "вниз" включен. Происходит переход к REST. Если бит tx, Y-D сброшен (OFF), то значения х и ?-1 сохраняются на стеке и Флаг овниз" включается.

В процедуре REST. если Флаг "ВНИЗ" выключен, происходит переход к NEXT-P. ЕСЛИ бит (X.Y-1) установлен (OH), то Флаг "ВНИЗ" выключается. В процедуре NEXT_P, если х-координатз равна 0. под-арогранма переходит к RETR. х-ко-ордината уменьшается, и. если новый бит (X,Y) сброшен (OFF), происходит переход к PLOT. В ЛРОПР дуре RETB. X и Y-координаты извлекаются из стека. Если X и Y равны 25Ь. то - возврат G BASIC, т. к. заполнение области завер шеио. иначе подпрограмма возвращается К RIGHT.

Процедура SUI3R должна подсчитать адрес бита (X.Y) в памяти. В BASIC этот адрес будет:

16384 + INT(Z/8) * ЙЬб - (2 -
U « 1НТГ//8)) * 32 « 164 •
]HT<z/64) * JHT(Z/ei e •
INT(2/64)),
где Z .175-У

Пары регистров ВС и DE сохраняются на стеке. В аккумулятор засылается число 175 и из этого значения вычитается Y координата. Результат копируется в н-регистр. Затем HL совраляется на стеке. Пять левых битов аккумулятора устанавливаются в о. а затем к ним прибавляется 64. Результат копируется в с-регистр. при умножении на 256 получаем:

16384*256* (Z-84NTC7/8)).

В аккумулятор загружается Z. это значение делитгя на в, результат копируется в регистр В. этот результат - 1НТ<2/в). Уста новка трех крайних правых битов в о при ротации дает значение 8•]NT(Z/64), которое загружаете я в D-регистр.

В аккумулятор загружается 7. и б крайних правых битов выключают ся, что дает 64«INT(Z/64>. это значение загружается в Е регистр. Значение из С-регистра копируется в Н. в ЗККУНУЛЯТОР загружается X-коорлината, это значение делит ся на в. а результат копируется в L.

В аккумулятор затеи загружается значение Е-регистра и к нему прибавляется содержимое В. значение D-регистра вычитается и ре эультат загружается в DE, это значение умножается на 32, DE восстанавливается из стека и прибавляется к HL, Т. о. HL теперь ярапит адрес бита IX,V).

В аккумулятор загружается первоначальное значение х. Установка пяти левых битов в ноль дает зна чение х 8»1НТ(Х/в). в в регистр затеи загружается 8 МИНУС значение

аккумулятора, чтобы использовать его в качестве счетчика. Аккумулятор устанавливается в 1, и это умножается на 2 (B-1! раз.

в этот момент в аккумуляторе необходимо установить бит. который соответствует биту (X, У) с адресом в HL, DE и BC затем восстанавливаются из стека, и SUB8 выполняет возврат в ОСНОВНУЮ программу.

6. 9 построение шаблонов.

Длина: 196

Количество переменных: 2 Контрольная сунна: 20278 назначение: Эта арограина чертит шаблон любого размера на экране, под шаблоном понимается любая, ранее определенная фигура. Переменные:

Имя - x_start Длина -1 Адрес - 23296

комментарии: x-координата первого пиксела. Имя - Y^start Длина -1 Адрес - Z3397

Комментарий: Y-координата первого пиксела. Вызов программы:

RANDOMIZE USR адрес Контроль ошибок: Если строковая переменная, хра-няеая информацию по шаблону a$, не существует, инеет нулевую длину или не содержит никакой информации, программа возвращается непосредственно в BASIC. Это происходит также в случае, если Y_start больше, чем 175.

Комментарий: Это полезная программа для хранения Фигур в памяти и быстрого вычерчивания их на экране. Использование этой программы: (1) LET At- "информация по

шаблону' (11IPOKE 33296, X-координата

первого пиксела (lll)POKE 23297, Y-координата

первого пиксела (lv> RABDOHIZE use адрес

Информация шаблона - это символов. КОТОРЫЙ инеет следующий формат:

" 0 " поместить точку " 5 " уменьшить X-координату " б " уменьшить ?-координату " 7 " увеличить X-координату о B " увеличить У-координату Лобне другие символы

игнорируются

Программа включает в себя возможность 'wrap-round', т.е.. если X-координата выходит за левую часть экрана, шаблон появляется справа и т. о.

чтобы иэиеиенить программу для использования иной. строковой переменной вместо a$. НУЖНО изменить 65* (КОД 6УКВЫ A) На КОД иного символа.

ЛИСТИНГ ПАПИНЫХ КОДОВ НЕТХА АССЕМБЛЕР ЧИСЛА ДЛЯ ВЕСЛА

```
            LD HL,(23627) 43 75 92
8EXT_V            LD A, (HL) 136
                        CP 128 254 128
KET Z 200
BIT 7,A 203 127
            JR HZ.FORHXT 32 23
                        CP 96 254 96
            JR HC, HUHBER 48 11
                        CP 65 254 65<
            JR 2.FOUND 40 35
STB I KG          INC HL 35
            LD E. (HL) 94
            INC HL 35
            LD D, (HL) 85
            ADD           ADD HL, DE 25
            JR            INCR 24 5
HUHBEE            INC HL 35
            INC HL 30
            INC HL 35
            INC HL 35
            INC HL 35
            INCR          INC HL 35
            JR NEXT_V 24 225
FORHXT                                    CP 224 254 224
JK C.B.BIT 56 5
            LD DE, 18 17 16 0
JB          ADD 24 236
```

```
H_BIT BIT 5, A 203 111
            JR Z, STRING 40 228
NEXT.B            INC HL 35
BIT 7. (HL) 203 126
            JR Z,NEXT_B 40 251
            JR NUMBER 24 228
FOUHD       INC HL 35
            LD C,(HL) 78
            INC HL 35
            LD B,4(HL) 70
            INC HL 35
            EX DE, HL 235
            LD A, (23297) 58 1 91
                                    CP 176 254 176
            RET HC 208
AGAIH       LD HL, (23296) 42 0 91
            LD A. B 120
            ORC 177
BET Z 200
            DEC BC 11
            LD A, IDE) 36
            INC DE 19
                                    CP 48 254 48
            JR HZ.HOT.PL 32 78
            PUSH BC 197
            PUSH DE 213
            LD A,175 62 175
            SUB H 148
            LD H. A 103
            PUSH HL 229
            AND 7 330 7
            ADD A. 64 198 64
            LD C,A 79
            LD A, H 124
                        RRA 203 31
BRA 203 31
                        RRA 203 3f
            AND 31 230 31
            LD B. A 71
            AND 24 230 24
            LD D.A 87
            LD A, H 124
            AND 192 230 192
            LD E. A 95
            LD H. C 97
            LD A, L 125
ISA. 203 31
ЯКА 203 31
                        RRA 203 31
            AND 31 230 31
            LD L. A 111
            LD A,E 133
            ADD A. B 128
SI» D 146
            LD E.A 95
-           LD D, 0 22 0
            PUSH HL 229
            PUSH DE 213
            POP HL 225
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL,HL 41
TOP DE 207
```

```
            ADD HL, DE 25
            POP DE 209
            LD A, E 123
            AND T 230 7
            LD B. A 71
            LD A, 8 62 8
SOB B 144
            LD B. A 71
            LD A,L 62 1
ROTATE            ADD A. A 135
            DJNZ ROTATE 16 253
                        RRA 203 31
            POP DE 209
            POP BC 193
OK (HL) 182
            LD (HL).A 119
HERE JK AGAIH 24 165
HOT.PL                                      CP 53 254 53
            JR HZ, DOVK 32 1
            DEC 1 45
DOWH                                  CP 54 254 54
            JR BZ. UP . 32 8
            DEC H 37
            LD H. A 124
                                      CP 255 254 255
            JR HZ. SAVE 33 19
            LD H. 175 38 175
UP                                    CP 55 254 55
            JR HZ, RIGHT 32 8
            INC H 36
            LD A, H 124
                                      CP 176 354 176
            JR HZ, SAVE 32 7
            LD H, 0 38 0
RIGHT                                 CP 56 254 56
            JR HZ. SAVE 32 1
            INC 1 44
SAVE        LD (33296). HL 34 0 91
            JR HERE 34 215
```

Как она работает:

Для нахождения адреса строковой переменной используется немного иэменеавая первая часть про-граммы 'Поиск подстроки'.

длина строковой переменной загружается в BC, а адрес первого символа a$ загружается в DE, в аккумуляторе устанавливается начальное значение V. и. если оно больше 175, подпрограмма возвращается в BASIC, в н-регистр загружается Y-координата. а в L -X-координата. Бели значение пары регистров BC равно о, подпрограмма возвращается в BASIC, т. к. достигнут ковеп строкОВОЙ переменной. BC уненьшается, чтобы по-

казать, что обрабатывается следу-пяий символ, следующий символ загружается в аккумулятор и DE увеличивается, указывая на следующий байт. Если аккумулятор не содержит код 48, ПРОИСХОДИТ переход к IIOT_PL. Точка (X.У) вычерчивается, используя процедуру SUBR из программы "Закрашивание контура". Затеи программа возвращается назад - к 'AGAIN'.

В процедуре HOT_PL. если акку-НУДЯТОР содержит число 53, X-ко-ордината уменьшается, в процедуре DOVH, если аккумулятор не содержит число 54. делается переход к UP. v-координата уменьшается, и. если ее значение становится равным -1, ?-координата устанавливается на значение 175.

В процедуре UP. если аккумуля-ТОР не содержит 55, происходит перекод к RIGHT. Y-координата увеличивается, и. если она равна 176, то Y-координата устанавливается в О. В процедуре EIGHT, если аккумулятор содержит значение 56, X-координата увеличиваете я. В процедуре SAVE координаты x и V помешаются в память, а программа делает переход j^ffiRE.

6.10 Увеличение экрана и копирование.

Длина: 335

Количество переменных: в

Контрольная сумма: ЗЗббЗ

Назначение: Эта программа копирует часть дисплея в ДРУГУЮ область экрана, увеличивая копию по X или со V.

Переменные: см. РИС. 3

имя длина	Адрес	Комментарий
upper _Y_co-ord 1	23296	Y-координата верхнего ряда
lower_Y_co-ord 1	23297	Y-координата нижнего ряда
rlght_X_co-ord 1	2329в	X-координата крайней правой колонки
left_X_co-ord 1	23299	X -координата крайней левой колонки
horizontal-scale l	23300	увеличение по 4:
vertical-scale 1	23301	увеличение по У
new_left_co-ord 1	23302	x-координата крайней левой колонки области, в которую делается копирование
new_iower_co-ord 1	23303	Y-координата низшего ряда области, в которую делает-копирование

вызов программы:

RAHDOKIZE USR адрес Контроль ошибок: Программа возвращается в BASIC, если одно из следушиж условий верво:

(1) borlzontal_scale=o (111 vertlcal_scale=o till) upper_Y_co-ord больше.

чем 175 (lv) nev_lower_co-ord больше.

чем 175

(v) lower_Y_co-ord больше, чем upper_Y_co-ord (si) left_X-co-ord больше.

чем rieht_x_co-ord Однако, для краткости программы нет контроля, который проверял бы возможность размещения новой

картинки на экране, веди этого ие получается, может произойти сбой, программа также требует большого объема свободной области ОЗУ. и. если это не доступно, может ПРОИЗОЙТИ сбой.

комментарий: Эта программа - не перемешаемая из-за процедуры PLOT. Она размешается по адресу 65033 и.

Если скопированная область экрана имеет тот же самый размер, что и оригинал, масштаб должен быть установлен в 1. для двойного размера загружается масштаб г: 1. для тройного размера загружается масштаб 3:1 и т. д.

```
ЛИСТИНГ НАКИННЫХ КОДОВ НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD IX.23296 221 33 0 91
            LD A,175 62 175
                        CP (IX'0) J21 190 0
            RET C 216
                        CP (1X»7) 221 190 7
            RET C 216
            SUB A 15!
CF (1X*4) 221 190 4
            RET Z 200
                        CP (1X*5) 221 190 5
            RET Z 200
            LD HL,(23295) 42 0 91
            LD B, L 69
            LD A, L 125
            SUB H 148
            RET C 216
            LD (23298).A 50 0 91
            LD E. A 95
            LD HL, 123298) 42 2 91
            LD C,L 77
            LD A, L 125
            SUB H 148
```

```
            RET C 216
            LD 123298). A 50 2 91
            PUSH BC 197
            LD L. A 111
            LD H. 0 36 0
            INC HL 35
            PUSH HL 229
            POP BC 193
            INC E 28
            ADD          DEC E 29
            JR Z. REMAIM 40 3
            ADD HL, BC 9
JK          ADD 2* 250
REMAIH            LD A, L 125
            AND 15 230 15
            LD B. A 71
            POP HL 225
            LD C, L 77
            JR HZ. SAVE 32 2
FULL        LD B, 16 б 16
SAVE        PUSH HL 229
            CALL         SUBR 205 13 255
            AND (HL) 166
            JR Z, OFF 40 2
            LD A, 1 62 1
OFF         POP HL 225
                         RRA 203 31
RL E 203 19
RL D 203 18
            LD A, L 125
                                     CP (IX»3> 221 190 3
            JR Z, NEXT_R 40 6
            DEC L 45
H_BIT       DJNZ SAVE 16 231
            PUSH DE 213
            JR FULL 24 226
NEXT_R            LD L. C 105
            LD A, H 124
                                     CP (IX»1) J21 190 1
            JR Z, COPY 40 3
            DEC H 37
            JR N_BIT 24 241
COPY        PUSH DE 213
            LD B, 0 60
            LD H,B 96
LI) L. B 10Ч
RESET       LD (2 3306). HI. 34 10 91 I
            LD A. B 120 II
            ORA 183 I
            JR HZ.              RETR 32 3 fl
            POP DE 209 I
1,D B. 16 616 II
            RETR         SUB A 151 II
            DEC B 5 II
RR D 203 26 I
RR E 203 27 I
RL A 203 23 I
            PUSH DE 213 I
            PUSH BC 197 II
            PUSH AF 245 II
            LD H. 1 38 1
LOOP        LD L.1 46 1
PRESER            LD (23304). HL 34 B 91
            LD A, (233071 1)8 11 91
            LD HL,0 33 0 0
```

```
            LD DE, (23301) 237 91 5 91
            LD D. L 85
HULT1P            ORA 183 t
            JR Z.CALC 40 6 1
            ADD HL, DE 2b II
            DEC A 61 II
            JR HU1.TIP 24 249 It
L.JUHP            JR RESET 24 208 II
CALC        LD A,(23303] 5» 7 91
            ADD A.L 133 LK HL,(233041 42 и 91
            ADD A. L 133
            DEC A 61
            PUSH AF 245
            LD A, (23306) 58 10 91
            LD HL, 0 33 0 0
            LD DE,(23300) 237 91 4 91
            LD D.L 85
REPEAT            ORA 183
            JR Z.COHTIH 40 4
            ADD HL,DE 25
            DEC A 61
            JR REPEAT 24 249
COHTIH            LD A, (2330E) 58 6 91
            ADD A.L 133
            LD HL,(23305) 42 9 91
            ADD H.L 133
            DEC A 61
            LD L.A 111
            POP AF 241
            LD H.A 103
            POP AF 241
            PUSH AF 245
            ORA 183
            JR HZ, PLOT 33 7
            CALL SUBR 205 13 255
CPL 47
            AND (HL> 166
            JR POKE 24 4
PLOT        CALL         SUBR 205 13 255
            OR(HL) 162
POKE        LD <HL).A 119
            LD HL,(23304) 42 8 91
            INC L 44
            LD A,(23301) 58 5 91
            INC A 60
                        CP L 189
            JR HZ.PRESER 32 165
            INC H 36
            LD A-(23300) 58 4 91
            INC A 60
                        CP H 168
            JR HZ,LOOP 32 155
            POP AF 341
            POP BC 193
            POP DE 209
            LD HL,(23306) 4E 10 91
            INC L 44
            LD A,(23298) 58 2 91
            INC A 60
                                    CP L 189
            JR HZ, L_JUMP 32 164
            LD L.0 46 0
            INC H 36
            LD A, (23296) 58 0 91
            INC A 60
```

```
                                CP H 188
            JR HZ, L_ JUMP 32 154
            RET 201
            SUBR          PUSH BC 197
            PUSH DE 213
            LD A,175 62 175
            SUB H 148
            LD H-A 103
            PUSH HL 229
            AND 7 230 7
            ADD A,64 198 64
            LD C, A 79
            LD A-H 124
            RRA 203 31
            RRA 203 31
            RRA 203 31
            AND 31 230 31
            LD B. A 71
            AND 24 230 24
            LD D.A 87
            LD A, H 124
            AND 192 230 192
            LD E. A 95
            LD H,C 97
            LD A,L 135
            RRA 203 31
            RRA 20J 31
            RRA 203 31
            AND 31 230 31
            LD L.A 111
            LD A-E 123
            ADD A. B 128
            SUB D 146
            LD E.A 95
            LD P. O 22 O
            PUSH HL 2г9
            PUSH DE 213
            POP HL 225
            ADD HL, HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL,HL 41
            ADD HL, HL 41
            POP DE 209
            ADD HL, BE 25
            POP Ш 209
LB A.E 123
            AND 7 230 7
            LD B,A 71
            LD A. в Ъ2 8
            SUB B 144
            LD B,A 71
            LD A, 1 62 1
ROTATE            ADD A, A 135
            DJNZ ROTATE 16 253
                        RRA 203 31
            POP DE 209
            POP BC 193
            RET 201
```

как она работает:

в IX загружается адрес буфера принтера Аля использования его в качестве указателя переменных. Если верхняя Y-координата или новая нижняя координата больше 175, программа возвращается в BASIC. Если значение увеличения по горизонтали или вертикали равно О. происходит возврат в BASIC.

В H-регистр загружается нижняя у-координата, а в L-регистр -верхняя У-координата. L-регистр копируется в B-регистр и в аккумулятор. H-регистр вычитается из аккумулятора и подпрограмма возвращается в BASIC, если результат отрицательный.

Значение аккумулятора затем помещается в ячейку 23298 для использования в качестве счетчика. Пара регистров BC затем сохраняется на стеке.

Регистр HL загружается значением аккумулятора, увеличивается и копируется в регистр BC, BC прибавляется к HL E раз. результирующее значение в HL является числом пикселей для копирования.

B аккумулятор загружается значение L-регистра и 4 крайних левых бита устанавливаются в о. Результат копируется в B-регистр для использования его в качестве счетчика.

Пара регистров HL восстанавливается из стека и регистр L копируется в с-регистр. Если B-регистр содержит О. в регистр загружается число 1б - это количество битов в регистровой паре. Затем вызывается процедура SUBR и в аккумулятор загружается значение РО1ЯТ (L,H). Пара регистров DE сдвигается влево, а значение бита из аккумулятора загружается в крайний правый бит E-регистра.

Если L регистр равен левой X-координаате, подпрограмма переходит к HBXT.R (следующий РЯД). Иначе уменьшается L-регистр, а затем - B-регистр. Если B-регистр не содержит О. подпрограмма возвращается к SAVE для аодачн следующего бита в пару регистров DE, ЕСЛИ в-регистр содержит О. пара РЕГИСТРОВ DE помещается е стек и ПРОИСХОДИТ переход к FULL.

В процедуре NEXT_R в L-регистр загружается правая X-координата. а в аккумулятор загружается значение н-регнстра. Если значение аккумулятора равно нижней Y-координате, ПРОНСгодят переход к СОР?, т. к. последний пиксель для кодирования подан в DE, Нваче. к-регистр уменьшаете я. указывая на следующий ряд. и подпрограмма возвращается к H_B1T.

в процедуре СОРУ содержимое пары DE помешается в.стек, а B. H и L-регистры устанавливается в 0 для использования юс в качестве

счетчиков. Содержимое пары регистров HL помещается в ячейку с адресами 23306/7 - HL теперь может использоваться как счетчик для последующих циклов без использования стека. Если B-регистр содержит о, DE восстанавливается из стека, а регистр B повторно устанавливается на значение 16. определяя количество пикселей, хранимых в DE, B-регистр уменьшается, показывая, что бит информации удален из DE, Крайний правый бит E-регистра загружается в аккумулятор, а пара регистров DE сдвигается вправо. DE, BC и AF поне-ааются в стек до тех пор. пока не выполнятся определенные расчеты.

B регистры H и L загружается 1 для использования их в качестве счетчика, а HL помешается в ячейки с адресами 23304/5. в аккуну-лятор загружается значение байта по адресу 23307 - это один из счетчиков, сохраненных ранее, в пару регистров DE загружается масвтаб по вертикали. Это значение затем умножается на значение аккумулятора, а результат подается в HL, Это значение прибавляется к новой нижней V-координате в аккумуляторе. Байт по адресу 23304 затеи добавляется к акиунулятору. а резуллЩт уменьшается.

АККУМУЛЯТОР теперь хранит V-координату для построения следующего пиксела. Это значение хранится в стеке до тех пор, пока подсчитывается X-координата похожим способом. После расчета X-координата загружается в L-регистр. Y-координата восстанавливается из стека и загружается в н-регистр. B аккумуляторе устанавливается последнее значение, хранящееся в стеке. Если оно равно i, то должна быть построена точка (X.У1. иначе должна быть выполнена процедура UHPLOTTED. Вызывается SUBR и выполняются соответствующие действия.

Пара регистров HL загружается счетчиками цикла, хранящимися по адресам 23304/5. Регистр L увеличивается, и, если он не содержит значение < 1 -«vert lea I .scale), подпрограмма возвращается к PRESE8. Регистр и увеличивается, н, если он не содержит значение (I+horizontal.scale). происходят переход к LOOP.

лары регистров AF,BC и DE восстанавливаются из стека, а в пару регистров HL загружается второе значение набора счетчиков никла, которое хранится по адресам

233О6/7. регистр L увеличивается, и происходит переход к RESET, если результат не равен спвМ_X_ co-ord - le-f t_X_co-ord * 1).

Регистр L устанавливается в о - это первоначальное значение счетчака цикла. H-регистр затеи увеличивается и подпрограмма переходят к RESET, если результат не равен (upper_Y_co-ord - lower. ?_co-ord + п. Программа возвращается В BASIC.

процедура SUBR идентична той. что используется в программе 'Закрашивание контура'.

ПРОДОЛЖЕНИЕ СЛЕДУЕТ

Алексеев А. Г.

МАСТЕРФАЙЛ-09 полная русификация

Окончание, начало см. zx-РЕНО-92, СТР. 29~зг.

Текстовые сообаеяия. выводимые на экран в тобой программе кроме печатаемых символов содержат такие управляющие коды AT, TAB. IKE. PAPER и т. д. Их без крайней необходимости лучше не изменять, так как это может нарушить работу программы, но в некоторых случаях придется менять и их. Для этого в программе-мониторе предусмотрена строка 331, Если Вы уберете KEM из строки 231 и подставите REM Б начало строки 230. то работа программы несколько изменится. Будет такой же вывод на экран, так же запрашивается информация, но теперь без кавычек, это означает, что ожидается ввод кода числа от 0 до г5_5. которое после нажатия "ENTER" непосредственно будет записано в нанять. Режим ввода кодов будет нужен редко, но все-таки иногда ПРИГОДИТСЯ. Для того, чтобы остановить ПРОГРЗННУ в этой режиме, надо просто нажать "курсор ВНИЗ" (CAPS SHIFT+6) или нажать "STOP" (SYMBOL SHIFT * A) и "ENTER".

теперь подставьте REM в обе строки 230 и 231 и сделайте RUN. После ввода адреса Вам будет выдан дамп памяти. Этот режим Вы будете использовать для поиска текстовых сообщений в программе. С этого момента можно начинать работу, но прежде - несколько примеров.

Введите адрес 60350. вы увидите на экране:

```
60350 237 GO SUB
60351 225 LLIST
60352 aoi <>
60353 22 ?
60354 2 ?
60355 O ?
60356 17 ?
60357 б ?
60358 86 V
60359 69 E
60360 82 R
60361 84 T
60362 73 I
60363 67 C
60364 65 A
60365 76 L
60366 зг
60367 76 L
60368 73 I
60369 78 H
60370 69 E
60371 255 COPT
```

в ячейках 6035О... 60352 находится какая-то программа в машинных кодах, а вот с ячейки 6О353 начина-

ется строка синвотов. при этом в ячейках 60353... 60357 находятся управлявшие символы, а сан текст расположен в ячейках 60358. .. &0370. Потом идет символ с кодом 2S5. КОТОРЫЙ завершает строку текста. Далее опять начинается программа в машинных кодах.

Управляющие символы имеют еледуюшее значение:

код: 16 - УПР. IKE

код: 17 - УПР. PAPER

код: 18 - УПР. FLASH

код: 19 - УПР. BRIGHT

код: 20 - УПР. INVERSE

КОД: 21 - УПР. OVER

код: 2E - УПР. AT

код: 23 - УПР. TAB Так что комбинация кодов в ячейках 60353... 60358 эквивалентна фрагменту Бейсик-строки:

... AT 2; O; PAPER &; ...

В ванном случае вас интересует только текст. КОТОРЫЙ должен быть заменен:

VERTICAL LIHE

ВБРТИК. ЛИНИЯ

(Ножете. убрав REM из строки 230. запустить программу, задать адрес 60358 и попробовать заменить текст на РУССКИЙ.)

ДРУГОЙ пример. После старта программы-монитора задайте адрес 58425. В дамне памяти Вы узнаете текст основного неню HF 09:

A ADD A RECORD. . .

CCH00SE A REPORT

DDISPLAY/PRINT. .

Правда он при работе программы выводится несколько иначе:

ADD A RECORD. ..... A

CH00SE A REPORT. ... C

DISPLAY/PRINT. ..... D

В памяти сначала идет символ той клавиши, которая должна быть нажата, затем текст сообщения. Вдаваться в детали работы программы нет необходимости. V нас другая задача, оставив без изменения символы тех клавишей, которые должны нажинаться (первые буквы, они набрав» в режиме курсора (C1). надо заменить текст на русский:

АНОВАЯ ЗАПИСЬ. . .

СОБЗОР ФОРМАТОВ.

ШШСПЛЕН/ ПЕЧАТЬ.

Еве пример, после старта монитора задайте адрес 59759. На экране увидите:

```
59759 78 8
59760 79 O
59761 32
59762 84 T
59763 73 I
59764 84 T
59765 76 L
59T66 197 OR
```

При работе программы эта надпись выводится так:

HO TITLE

Откуда же берется последняя буква <E>?

Вообше надо сказать, что перед тем. как в машинных кодак подана команда вывода строки символов на экран, предварительно задан адрес начальной ячейки, где расположена строка символов и длина этой строки. Анализ программы HF 09 показывает, что здесь применяется еще и другой способ вывода, длина выводимой строки не указывается, однако во время вывода анализиру ется код выводимого символа. Если это код с 0 по 137, то продолжа ется вывод на экран, а если код символа больше или равен 128. то это значит. что процедура вывода этим символом заканчивается, при этой на экран выводится символ, код которого на 128 меньше. чен содержащийся в памяти. в ячейке 5976б стоит OR. Его код (читаем на экране) равен 197. 197-188 69. По таблице колон "Спектрума". а если ее нет под рукой, то сделав PRINT CHRS 69. выясняем, что это ну коду соответствует буква "к" Вот откуда взялась последняя бук вз.

Этот текст можно заменить еле дующим образом:

но TITLE не назв.

При этой буквы "не назв" зане няен непосредственно русскими буквами, а вместо точки, код ко-торой (выясняем: PRINT CODE ". ") равен 46. надо ввести символ с кодом 46 + 1гвМ74. По таблице кодов "Спектрума" или сделав PRINT CHR$ 174. определяем, что это VAL$. Пожно ввести его в тон же режиме, нажав "EXT. HODE". затеи •SYHBOL SHIFT-+-J". Часто придет ся вводить в качестве последнего символа - "пробел" (его код 32>. Это будет символ, код которого равен 32 * * 28 =160. По таблице кодов находим, что это символ "о" UDG-графики, вводя его, нажмите -GRAPH' (CAPS SHIFT+9). за-

тем "Q", затеи еще раз "GRAPH" (пусть вас не силвает, что напечатается что-то непонятное, так как на месте символов шзс-гра>ики находятся коды программы HF 09). однако- могут попадаться такие символы, которые с клавиатуры вводятся в режиме курсора [к]. Например, начиная с адреса 64522 наводится текст:

645й2 65 а
64523 82 R
64524 71 G
64525 32
&4э2б 76 N
64527 79 о
64528 S4 т
64529 32
64530 78 н
I 64531 S5 U
I 64532 77 н
I 64533 69 е
I 64534 62 R
64535 73 I
64536 195 нот

в ячейке 64536 "спрятана" буква. код которой 195-128=67. это

латинская буква "с", заменяем эту строку на русский текст:

AKG NOT NUMERIC арг. не числовой

при этой вместо последней русской буквы "и", (ее код выясняем, сделав PRINT CODE "и" - при этом "и" набрана в режиме курсора [D. так как включен русско-ла-тинский символьный набор; он равен 1о6) надо ввести символ с кодом 106*126=234. Это REM. Ввести этот символ, находясь в режиме курсора 11.) или [с] никак не удастся. Для этого надо переключить курсор на [к1 (командный режим». Как это сделать? Введите ключевое слово "THEN" (SYMBOL SHIFT+G). Теперь курсор стал [к] и вы можете ввести "вен", нажав "е". теперь нажните "курсор влево" и удалите "THEM" при помощи -DELETE" (CAPS SHIFT+0). Далее -"ENTER" - ввод кода в нанять.

Можно это сделать и иначе -просто остановить программу и "вручную" сделать роке 64536,234.

Теперь вы знаете, как располагаются текстовые сообщения в программе HF 09 и умеете работать с программой-монитором. Подставьте йен в обе строки 230 и 231 и начинайте работу. Запустив программу -монитор, задайте адрес 57326. (с адреса 5&56о по 57327 расположен символьный набор, тан делать нечего. > сейчас задача -просмотреть все коды программы от начала до конца, найти все текстовые сообвеиия в программе и записать их адреса и сани сообвевия на бумагу, оставляя место для последующего перевода.

Далее, вооружившись англо-русским словарей, начинаем пе-

ревод, не надо стремиться переводить текст дословно, так как мы слишком силино сжаты ранками того места, которое отведено в программе под ту или иную Фразу. Более подойдет литеретурный перевод. при этом, конечно, лучше, если вы уже достаточно поработали с программой HF 09 и Вам понятен смысл переводимых сообщений. Это позволит Вам найти подходящую по смыслу замену, даже если она не является переводом английской фразы, что касается программы HF 09, то сообщения, которые Вы тан встретите

и их перевод, реализованный в ноем варианте "HF09 RUS". приведен ниже. Когда же Вы возьметесь за какую-нибудь другую программу, то вам придется проделать самим всю эту работу, это. пожалуй, самая большая по затратам времени. ответственная и творческая часть работы. Так что запаситесь терпением, может быть не на один день, остальное - дело чисто механическое.

текстовые сообщения программы HF 09 и их перевод

вначале каждого сообщения указан адрес, с которого необходимо произвести замену текста. Далее -английский текст и под ним

русский вариант перевода.

подчеркивающая черта обозначает "пробел".

символы, коды который должны

быть на 120 больше, отмечены "*" в конце строки, далее идет код символа и сан символ или ключевое слово.

57764:
ITEH_ALREADY-.IH_RECORD « 196 BIN
это_поле_уже_введеш__ » 160
Q-GRAPH 57787: AADD_ITEH. ..... RREPLACE_ITEM. . .
АНОВОЕ_ПОЛЕ. .... R ИЗМЕНЕНИЕ ПОЛЯ.
EERASE.ITEM. .... HNEXT_ITEM. .....
ЕУДАЛЕНИЕ_ПОЛЯ. . НВЬБОР_ПОЛЯ. ....
DDISPLAY/PRINT. . GGET_ITEH. ......
ОДИСПЛЕЙ/ПЕЧАТЬ. СВЫЗОВ_ПОЛЯ. ....
PPROHPT^ITEMS. . . -AHOTHER_RECORD. РАВТОЗАПРОС. .... = СЛЕДУЮЩАЯ ЗАП. .
ИИА1Н НЕКТJ .....
нглАвноЕ.к&яо. ..
57940:
EfITKR_TEXT_I-128_CHARS. ВВОЯ_ТЕКСТА_1-12в_СЙМВ.
57988:
GIVE-DATA-RSF « 198 AHD
METKA. ДАННЫХ. • 160 Q-GRAPH
58089: HOT.HAHED
НЕТ^ИМЕНИ
58130:
SAVE „PROG/FILE,
ЗАПИСЬ^ПР. /ФАЙЛ
58243:
7-TO_COKFIRH » 205 STEP
У-ЕСЛИ_ДА__ * 1&0 Q-GKAPH
58425:
AADD^A,RECORD. . . CCH00SE.A^REPORT
АНОВАЯ_ЗАПИСЬ. . . СОБЗОР_ФОРМАТОВ.
DDISPLAY/PRINT. . EEDIT-FORMAT_DEF ШШСПЛЕИ/ ПЕЧАТЬ. ЕРЕДАКТ. „ФОРМАТА
LLOAD_A_FILE. . . . NHAffiLDATA^REF. . ЬЗАГРУЗКА„ФАЙЛА. НИМЕНА. ПОЛЕЙ ДАЛ
SSEARCH_THE-FILEIIHVERT.SELECTH. SПОИСК- ..... IИНВЕРСИИ.ВЫБОРА
RRESET_SELECTIOHFPURGE_SEL_RECDS ЙСБРОС_ ВЫБОРА. . . РУДАЛЕН. ВЫБР. ЗАП
TTOTAL/AVERAGE. . VSAVE_PROG/FILE. ТИТОГ_ОБЖ. /СРЕДНУЗАПИСЬ_ПР. /ФАЙЛ
UEXEC.USER.BASIC
ивыполнить_ BASIC
58903:
FILE_OHLY. ..... F
ТОЛЬКО_ДАННЫЕ. ..... F
58998:
_RECS = 00000_SEL--00000_SPA=O0000_
... ЗАП = OO000_ВЫБ^OO000_CBE:OO000_

59326: 59759:
FILE_FULL NO_TITLE » 197 OR
НЕТ_МЕСТА НЕ.НАЗВ. " 174 VAL$
60065: 60077:
DELETE СОРУ
УДАЛ. _ КОП.
60082:
ALREADY_DEFIHED » 196 BIH
МЕТКА_ЗАНЯТА__ « 160 Q-GRAPH
60О97:
НО_SUCH_FORHAT_DEF1HED * 196 BIH
ТАКОЙ.ФОРМАТ_НЕ_ЗАДАН. » 160 Q-GR
601124: 60287: 60300:
REF_0 GEHERAL SEQU__:
МЕТ. О ГЛАВНЫЙ СОРТ. _:
60312: 60325:
BORDER _INTVL 002
.БОРДЮР ЙНТЕРВ002
60358: 60367:
VERTICAL. LIKE LIJDS_ACROSS
ВЕРТНК. .ЛИНИЯ ГОРИЗ. ЛИНИЯ
60435: 60495: 60574:
BOX TEXT DATA_REF__:
ПР. ТИС ПОЛЕ_ДАН. _:
(Примечание: первод слова TEXT как TEXC - не очень удачный вари-
ант, но мы ограничен» нестон в памяти. То же касается и перевода слова BOX - ПР. [ЯНОУГОЛЬНИК}. Как с этим бороться, мы еще рассмотрим ниже.)
60591: 60603:
WIDTH.000 DEPTH_000
гаРИНА000 BUCOTAO00
60615: 60645:
TOLL: PAPER.
НОЛЬ: БУМАГА
60677:
GIVE.REPORT.REF • 198 AHD
НЕТКА_ФОРНАТА__ > 160 Q-GRAPH
60793: 60801: 60809:
IHY.H BRI.Y PAD_V
ИНВ. Н ЯРК. Y ФОН.У
60817: 60828: 60872:
FLASH..H HPRT-42 LIHE.O00
НИГАН. S СИНВ=42 .СТР. 000
6О883: 60931:
COL.O00 X,C00RD_000__LEHGTH_000
КОЛ. 000 X-КООРД. 000__ДЛИНА_000
60957: 61205: 61213:
y.C00RD.O00 LEHGTH: WIDTH=
У-КООРД 000 ДЛИНА._ 1ИРИНА
61220: 61227: 61234:
DEPTH: PAD HULL. TEXT =
ВЫСОТА ФОН НОЛЬ.ТЕКСТ
61361: 61367: 61380:
LIHE: HICRO-PRT. 42_PITCH
СТР. ИИКРОПЕЧ . 42,СИМВ.

61392: 61457: 61467:
COLUHH: BRIGHT, INVERSE.
КОЛОНКА ЯРКОСТЬ ИНВЕРСИЯ
61478: 61484:
PAPER « 189 ABS FLASH.
БУМАГА « 225 LLIST НИГАН.
61516: 61525: 61596:
X.C00RD: Y.C00RD: BORDER:
X-КООРД. У КООРД. БОРДЮР.
61603: 61612:
SEQUEBCE: ' 189 ABS INTERVAL:
СОРТИРОВ. " 17» VAL» ИНТЕРВАЛ,
61639.
AADD.KEU.FORMAT. RREVIEV.FORMAT. .
ЛНОВЫН.ФОРНЛТ. . ВИЭНЕНЕНИЕ ФОРП.
MMAIH.MEHU. .....
НГЛЛВНОЕ-НЕНВ. . .
61688:
AADD.HEU.ELEHEHTRREPLACE.ELEHEHT АНОВЫЯ ЭЛЕМЕНТ КИЭНЕНЕННЕ.ЭЛЕИ.
EEKASE.ELENEXT. . NNEXT ELEMEUT . . ЕУДАЛЕННЕ_ЭЛЕК. НВЫБОР.ЭЛЕМЕНТА.
CCPPt.FORHAT. . . . XDELETE FORMAT . СКОРИРОВ. *ОРП. . ХУДАЛЕНИЕ_«ОРН. .
DDISPLAT/PRIBT. . «PREVIOUS..HEHU . ВДЖСП/ЖЯ/ПЕЧАТЬ. НПРЕДЫДУЩЕЕ.МБПО
61817:
DDATA.FROll.liECD. LLITERAL_TEXT. . . ПДАННВЕ. ..... ЬЗАГОЛОВОК. ТЕКСТ
BBOX. ..... HHORIZOHTAL.LIHE
ВПРЯНОУГОЛЬНИК. . НГОРИЗОНТ. .ЛИНИЯ
WERTICAI^LIHE. . MPREVIOUS_MEHU. . VBEPTHE. .ЛИНИЯ. . НПРЕДШППЕЕ.НЕНЮ
' 62102: 62127:
TOTAI.__ = _ AVERAGE. = _
ОБЩЕЕ__:_ СРЕДНЕЕ. :_
62194: 62219:
GIVE.FILE.HAME ___ERROR__
ИИЯ_ФАЯЛА____ __ОШИБКА__
63177: 63196: 63205:
REPORT. МЕНО . . .MORE
ФОРМАТ. МЕНЮ ...ЕЩЕ.
63225:
fl A fO LL_ fl S fO IHGLE.PAGE •
197 OR
fl A fO -BCE. fl S fO -ЭКРАН.. " 160 Э-GRAPH
(здесь fl и fO - управляете символы с кодами 18. 1 и 18. 0 -включение и выключение режима мигания)
63331:
HBEXT_PAGE. ..... *ADVAHCE_I 9_RCS
ЫСЛЕДУН»ИИ_ЭКРАН*ВПЕРЕД-НА.1-9. .
OBACE.OKE RECORDBBACS_TO 1ST.REC ОНАЗАД_НА_1. .... ВВОЭВР. .В_НАЧА/Ю
PPRIIIT. ..... UUPDATE_TOP REC.
РПЕЧАТЬ. ..... UH3HEH. _ВЕРХ.ЗАП
EERASE.TOP.REC. . 00HIT.TOP.RECORD ЕУДАЛЕН. ВЕРХ_ЗАПОИСКЛГЧ. ВЕРХ ЗАП
CCOPУ.TOP. RECORDSSEARCH..THE.FILE СКОПИР. ВЕРХ.ЗАПБПОИСК. .....
TTOTAL/AVERAGE. . RSELECT. REPORT. . ТИТОГ.ОБ». /СРЕДНХВЫБОР_»ОРНАТА. .
MMAШ.МЕНи ..... QOUIT.THIS _МЮШ. МГЛАВНОЕ. МЕНЮ. . . ОУДАЛИТЬ.НЕНЮ. . .
64522:
ARG_HOT_mjMERIC " 195 KOT

АРГ. НЕ.ЧИСЛОВОЙ • 234 REM
64555: 64570: 64580:
ARGUHEHT: ALL. SEL.
АРГУМЕНТ. ВСЕ. ВЫБР
64596: 64607:
СНА». ИНН..
ТЕКС. ЧЯСЛ.
64623:
ASELECT.FROM.ALLLSELECT.FliOH.SB АВЫЮР.ИЗ.ВСЕХ. . ЬВЫ5ОР_ЛЗ.ВЫЕРАН
DDISPLAT/PKIIIT. . НПА1И-НЕЯи. .... ОДИСПЛЕИ/ПЕЧАТЬ. НГЛАВНОЕ_МЕНЮ. . .
64686:
CCHARACTER. .... SHUHBERIC ......
СТЕКСТОВЫИ. ..... НЧИСЛОВОИ. ......
MPREVIOUS.MEHU. . МПРЕДЫДУЩЕЕ.НЕНЮ
64737:
EEQUAL.TO. ..... UUHEQUAL.TO. ....
ЕРАВНО_АРГУНЕНТУиНЕ_РАВНО_АРГ. . .
LLESS.THAH. ..... GGREATER.THAB. . .
LHEHbEE. ЧЕМ.АРГ. 6БО/1ЫШ. ЧЕМ.АРГ.
SSTRIHG.SEARCH. . MMAIB.MEHU. .....
ЗСКАНИРОВАНИЕ. . . ИГЛАШОЕ. МЕНЮ. . .
65210:
HOH-HUHERIC.DATA: fl S fO KIP.
НЕ_ЧИСЛ! ДЛИН, fl S fO -ДАЛЫЕ.
fl U fO PDATE • 197 OR fl U fO -СТОП " 240 LIST
65320:
PPROGRAM«FJLE. . . FFILE.ONLy. .....
РПРОГР. *.ФАИЛ. . . РТОЛЬКО ДАННЫЕ. .
65353: SAVE_KAME_? ИМЯ.ФАЙЛА.

После того, как перевод закон чен на бумаге, загружаем программу-монитор. Убираем REM из строки 23О и начинаем замену английских текстов на русские а соответствии со своими записями. Не забывайте периодически выгружать результаты через RUN 5.

Попутно следует сказать о том. I что не стоит экономить ленту. стирая старый вариант и записыа вая на него новый, это позволит Вам вернуться назад в том случае.

чайно "запортите" программу, ошибочно изменив содержимое памяти где нибудь в области машинных ко дов.

Когда эта часть работы будет выполнена и записан последний вариант "русского" кодового куска. надо проверить, как все будет работать в н*вом виде. запустите праграину MF 09 и. после загрузки блоков. "HF LOADER" и "HF 09 LEER" остановите магнитофон и вставьте кассету с записью "русского" кодового куска, загрузите его. после старта программы, если Вы асе сделали аккуратно. Вы можете васладиться результатами своего труда. Но это потом. А сейчас надо внимательно посмотреть, нет ли где-нибудь грамматических ошибок, которые практически неизбежны в результате такой больной работы, и вообще, все ля выполняется "ладно" и SOPOBO. нет

жн где-нибгдь 'шероховатостей' перевода и т. д.

Если ошибок нет. то в обшей можно считать работу выполненной, но присмотримся к работе програн-ив повнимательнее,

Ну например. вместо слова "BOX* - ПРЯМОУГОЛЬНИК - мы ввели

•ПР. • <сн. ячейку 60435). это выглядит не очень удачно на экране, но. с другой стороны, что еве можно разместить в выделенных для этого трех байтах текста? Аналогично, вместо 'TEXT" - мы ввели

•TEK* (ячейка 60495). Та же история - не хватает места. Что ТУТ можно сделать?

Вспомним, как в нашивных кодах осуществляется вывод на экран. Непосредственно

перед выводом где

•то должен быть задан адрес начала строки текста, надо найти этот адрес и изменить его. расположив альтернативные текстовые сообше-ния на новой месте. Вот как это может выглядеть.

Текстовая строка 'BOX" начинается фактически с адреса 6О430:

60430 22 ? • - AT 2:0;

60431 г ? 604 зг о ?

60433 17 ? - PAPER 6;

60434 б ?

60435 66 в -текст

60436 79 О

60437 86 X

60438 255 СОРУ - "КОНец"

вместо "ПР." желательно было бы разместить ну хотя бы:

AT г о PAPER & ПРЯНОУГ. "квяеп-

для такой строки надо 14 байтов оаняти. Аналогично со строкой "TEXT* - Фактическое начало - адрес 60490. Замена:

AT 2 0 PAPER 6 ТЕКСТ "конец"

для этой строки надо 11 байтов памяти. Всего для ДВУХ строк нам НУЖНЫ 25 байтов. Подумаем где вы их найти. В нашем случае есть свободное место вереД символьным набором (расположенным с адреса 56560)- Запустите программу-монитор (ВЕН - в строке 321} и с ад реса 56560-25^56535 осуществите ввод. Вначале пять пробелов (они НУЖНЫ, чтобы потом на их место поставить уврaвоямте коды AT и PAPER}. затем текст "ПРЯИОУГ. ". затем еве пробел. И сразу же далее: еще пять пробелов, затеи слово 'ТЕКСТ' и еще один оробел. Ввод закончитея ячейкой 56559. тепер* остановите программу, подставьте ЯЕН в строку 230 и удади-те ЙЕН из строки 231. Запустите Впвограмиу-мовитор: ео то аоо и Ненова с адреса 56535 введите еле* │душшие кош: 22. г. о. 17. б, за-

тем восемь раз просто нажните "ENTER". ничего не вводя, затем введите: 255. 22. г. О. 17, 6, еще 4 раза нажмите. -ЕмТБй". и, наконец, введите последнее цифру 355. ввод завершен на ячейке 56559.

теперь надо найти в программе те места, где указаны адреса старых текстовых сообщений и изменить их на новые. Старый адрес для сообщения -BOX" - 60430. для поиска подставим КЕМ в строки 230 и 231 и введен НОВУЮ строку:

210 IF (FEEE a$25&iPEEK(A+I» 060430 THEU GO TO 300

Запустите программу RUH, задайте адрес 56328 - это будет начало поиска - и можете ОТКИНУТЬСЯ на СПИНКУ кресла, пока компьютер не выдаст Вам необходимую информа цию. На экране напечатается:

60405 14 ?

Не нажимайте "BREAK", подождите, пока не будет просмотрена вся программа, может быть на этот адрес есть ссылки и в ДРУГИХ местах или это случайная комбинация чисел, никакого отношения не имеющая к процедуре вывода. Поиск закончится сообщением об ошибке; "B Integer out of ranee. 2io:i". Это значит, что вся память просмотрена.

Место в программе, где указывается на искомый адрес, оказалось единственным. тем проие для нас.

Теперь найдем адрес, где указано начало второго, интересуюне-го нас сообщения - "TEXT" - это 60490. Изменим в строке 21 о программы-монитора число 60430 на &О49О и GO TO 200. На экране появилось:

60458 74 J

Сообщение об ошибке. Поиск закончен. Этот адрес тоже единственный, теперь оба адреса надо заменить на новые. Для сообщения "ЙРЯИОУГ. " это будет адрес 56535. а для "ТЕКСТ" - 56549. Нладшие и старшие байты этих чисел находим, выполнив:

PR1HT 56535-25&"INT(56535/256) PRINT 1иТ(Э6535/г56) PRIMT 56549-ПГГС56М9/256> PRINT 18Т(56549/г56)

ПОЛУЧИМ, соответственно, числа 215. 220. 229. 22О. теперь ВЫПОЛНИМ:

POKE вО4О5,215 POCE 60406.220 POCE &О458. 229 POKE 60459. 220

Осталось готовую программу за-

писать на ленту, во прежде, чем сделать вин -5. вспомните, что ваша программа стала длиннее ва 25 байтов, это надо учесть, ПРОИЗВОДЯ изменения в строке б программы-монитора :

SAVE Я* CODE 56560,8976

надо заменить на

SAVE M* CODE 56535,9001

Теперь можете сделать ИШ 5. Надо также изменить программу- загрузчик "HF LOADER", заменив CLEAR 56559 на CLEAR 56534.

Попробуйте новый вариант программы. Вы согласны с тем, что она выиграла от такой замены?

Еще одна деталь. при запуске программы, когда появляетс я на экране главное меню, слева вверху на синен Фоне появляется надпись:

"РИЛЕ:HP 09 LEER"

Вместо РУССко-латинского слова "РИЛЕ" надо бы написать "ФАЯЛ", во вот беда: это слово мы не нашли в кодах программы HF 09. На самом деле все очень ПРОСТО. Его тан и нет. Надо искать в ДРУГОМ месте. Это слово находится ВНУТРИ Бейсик-программы HF 09 LEER. Чтобы убедиться в этом, сделайте следуюшее. остановите программу, нажав в главном меню "L". затем "курсор ВНИЗ" (CAPS SHIFT*&). Tr-нерь сделайте PRINT F*. Вы УВИДЬ те надписи, выводимые на экран в режиме главного меню. Слово "РИЛЕ" находится в Файле данных -Бейсик-неременной F*. Как ПОСТУПИТЬ в этом случае?

Запустите программу HF 09. Находясь в главном меню, нажмите "V". а затем "F" и, указав имя. например "LEER", запишите пустой файл на магнитофон. (Это как раз и будет переменная F».) дальше загрузите опять ПРОграмму-монитор (и коды HF 09). остановите ее и загрузите переменную F*. подав ПРЯМУЮ команду:

LOAD "LEER" DATA F* ()

Теперь надо помнить о том. что нельзя подавать команду RUH. а только GO TO 1, так как подав RUH, Вы уничтожите переменную F*. просматривая дамп памяти примерно с адреса 24400. вскоре вы встретите фразы: "NASTERFILE BEP 09" и " РЯЛЕ: LEER ". Замените их на "HASTBRFILE 8US 09" и "ФАЙЛ:LEER" обычным способом. Теперь остановите программу-монитор и запишите переменную F* на магнитофон ПРЯНОЙ командой:

SAVE "LEER" DATA F» {)

далее надо запустить программу ИГ 09 и. находясь в главном меню, нажать "L". Задайте имя "LEER" и загрузите с магнитофона изменен-

ный вариант переменной F$. После выхода в главное меню Вы увидите, правильно ли Вы все сделали. Теперь можете записать измененную Бейсик-программу MF 09, нажимая в главном меню "V" затем "P" и указав имя "MF 09 LEER".

Смотрим, что еще в программе не так. В режиме "дисплей" при просмотре записей в правом нижнем углу экрана на голубом фоне появляется надпись:

.._ЕЩЕ_ или No_ЕЩЕ_ ("_" - обозначает пробел)

При этом в памяти постоянно хранится только запись "_ЕЩЕ_", а "No" или ".." подставляются программой непосредственно в строку, подлежащую выводу, в зависимости от алгоритма работы. Таким образом, если мы хотим сделать перевод текста, то на место английского "No" не помещается даже русское "НЕТ". А русско-латинская фраза "No ЕЩЕ" выглядит довольно-таки неудачно. В этом случае можно попробовать использовать символы, например "+" и "-".

Анализируя работу программы MF 09 (кстати в этом мне очень помог трехтомник "Программирование в машинных кодах", написанный "ИНФОРКОМОМ"), я нашел, как мне кажется, приемлемый вариант замены. Для реализации этого варианта надо сделать:

```
POKE 62759, 230   POKE 63155, 32
POKE 63037, 230   POKE 63205, 32
POKE 63123, 230   POKE 62206, 32
POKE 63146, 230   POKE 62757, 45
POKE 63121, 43    POKE 63040, 45
```

Теперь, если при просмотре записей в режиме "дисплей" есть еще записи, то в левом нижнем углу будет: "_+_ЕЩЕ_", а если больше нет записей, то: "_-_ЕЩЕ_". По моему, это выглядят более гармонично.

Теперь еще об одной существенной детали. При вводе данных, для переноса строки в режиме "дисплей", применяется символ "вертикальная линия" - С.В.Л. (это в режиме "EXT. MODE" нажимаем SYMBOL SHIFT + S). Код этого символа равен 124. Но в кодах ASCII в символьном наборе КОИ-7 "НС", который применен в нашем варианте MF O9, коду 124 соответствует буква "Э". Поэтому, как только в тексте встретится буква "Э", она напечатана не будет, а произойдет перенос текста на новую строку.

Устранить можно и эту проблему, заменив символ, дающий команду переноса на какой-нибудь другой. Вопрос только: на какой? Ведь любой другой символ может встретиться в файлах данных, введенных на нерусифицированной программе MF 09.

Я считаю, что мне удалось найти удачное решение. Сделайте:

```
POKE 58237,0      POKE 63812,0
```

Теперь вместо С.В.Л. для переноса строки будет использоваться любой символ, с кодом больше 127. Это любое ключевое слово, например "STOP" или "AT" или любое другое, которое Вам удобнее набирать. При этом в файл данных будет занесен код 0 (независимо от того, какое ключевое слово Вы ввели). Символ с кодом 0 - это в кодах ASCII управляющий символ "NUL", он для печати не используется. Но в режиме "дисплей" теперь по этому символу произойдет перенос на новую строку. Если же Вы при помощи переделанной программы будете просматривать файлы данных, введенные на непеределанной программе, то все символы переноса строки, находящиеся в тексте, станут видимыми.

Если при пользовании русифицированной программой MF 09 возникнут какие-либо неудобства, связанные с переносом строки, то напишите в ИНФОРКОМ, я готов придумать что можно сделать. (Например, можно написать несложную программу обработки уже готовых файлов данных, заменяющую все встречающиеся в тексте С.В.Л. на управляющие символы "NUL".)

Многие проблемы, существенные и несущественные, связанные с русификацией MF 09, мы решили. Но осталась еще одна очень важная проблема, пока еще нерешенная. Это сортировка по русскому алфавиту. Как производится сортировка по латинскому алфавиту? Очень просто. В кодах ASCII латинские буквы уже расположены в алфавитном порядке. Значит, располагая записи по возрастанию кодов, можно производить сортировку. С точки зрения программирования это довольно простая задача. С русским алфавитом сложнее. Здесь нет никакой связи между алфавитом и кодом символа. Поэтому сортировка по тому же принципу, что и латинского текста не годится. Причем проблема сортировки по русскому алфавиту гораздо шире, чем рамки программы MF 09. По-моему проблема вполне достойна того, чтобы вынести ее на общественный "Форум". Может быть, кому-нибудь из читателей удастся преодолеть ее? В таком случае заявите в ИНФОРКОМ. В свою очередь, я тоже пытаюсь найти приемлемый вариант решения этой задачи и когда закончу работу, сообщу об этом.

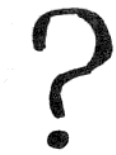
ЗАКЛЮЧЕНИЕ.

Мы закончили перевод программы MF 09 на русский язык, выполняя эту работу, Вы познакомились с некоторыми методами, которые могут применяться для этого и немного

"набили руку". Теперь Вы легко сможете применять "русский язык" при разработке своих программ.

Что же касается перевода фирменных программ, то тут следует отметить некоторые моменты. Приобретенные знания помогут Вам во многих случаях, однако будут встречаться разные "сюрпризы и хитрости", придуманные авторами (как было в MF 09, например, код последнего символа на 128 больше и др.). Процедуры вывода на экран могут быть самыми различными. Кроме того, трудности начнутся с загрузки программы, так как при нажатии "BREAK" программа "зависает" или происходит рестарт компьютера. Но даже если Вы удачно "взломаете" защиту загрузчиков программы, надо учесть, что очень многие коммерческие программы закодированы и поэтому, просматривая дамп памяти, мы ничего похожего на текстовые сообщения увидеть не можем. В этом случае надо, дизассемблируя программу, начиная со стартового адреса, найти процедуру декодирования, написать аналогичную процедуру в машинных кодах, раскодировать программу и только после этого заниматься переводом. Закодировать программу после перевода нет необходимости, проще изменить процедуру запуска, исключив элементы декодирования. Без знания машинных кодов Z-80 решить эту проблему, конечно невозможно.

* * *



"ИНФОРКОМ" получает массу писем от начинающих программистов с просьбами посоветовать им над чем бы стоило поработать, в какой области приложить свои усилия.

Мы считаем, что лучше всего в этой деле равняться на классиков. Поэтому пользуясь тем, что этот номер "ZX-РЕВЮ" является апрельским, мы поздравляем дорогих читателей с праздником 1-го апреля и рекомендуем обратить внимание на последние достижения всемирно известной фирмы PFA.

Мастеров "самого быстрого ксерокса" просим при перепечатке (или ином воспроизведении информации) не стесняться и смело указывать на первоисточник.

Проблем совместимости больше не существует.

Уникальная развлекательная программа разработана и выпущена фирмой PRESENT FROM AFRICA. Это первая версия имитатора "Super Real Bomb Simulator".

Программа отличается от всех известных аналогов тем, что ее не надо инициализировать, настраивать и вообще ее не надо загружать. Она не требует ни джойстика, ни клавиатуры.

Вам достаточно вставить кассету в магнитофон, нажать клавишу PLAY и спокойно ждать, когда она сработает.

Фирма особенно удовлетворена тем, что до сих пор от зарегистрированных пользователей не поступило ни одной жалобы (письма и заявления от родственников и друзей покойного фирма традиционно не рассматривает).

За особо дополнительную плату программа может поставляться на диске.

Решается вопрос о перспективах маркетинга программы в термоядерном исполнении (так называемый экспортный вариант). Это первая в мире программа, работающая на компьютере любой системы!

Проблем с продуктами больше нет!

Впервые в мире фирмой PRESENT FROM AFRICA выпущен имитатор продукта питания. Это завоевавшая всемирную известность программа "Shashlyik Basturma Simulator".

Программа состоит из двух частей. Первая часть - аркадная игра по нанизыванию кусочков баранины и лука на палочку.

Вторая часть требует специального блока питания (поставляется вместе с программой).

Подключив его и запустив программу, Вы уже через несколько минут сможете насладиться ароматом настоящего шашлыка, исходящим от Вашего компьютера.

Если запах немножко не тот, значит Вы забыли замочить компьютер в слабом растворе уксуса или в сухом вине. Сделать это лучше всего накануне вечером. Не забудьте добавить лука репчатого, перца молотого и посолить.

Суперновинка!

Развивая плодотворную идею программы "Shashlyik..." фирма PFA подготовила совершенно оригинальный программный продукт, объединяющий в себе абсолютно все известные жанры игровых, прикладных и вкусовых программ. Эта новинка поступила на прилавки под названием "Chuckie Egg F16".

Сначала Вы работаете в режиме "TOOLKIT", конструируя себе цыпленка-табака по вкусу. В этом режиме программа является также обучавшей по анатомии пернатых.

Увлекательный аркадный эффект возникает, если Вы ошибетесь и Ваш цыпленок начнет походить на утку или гуся - Вам придется отбиваться от группы сумасшедших яблок (APPLES), желающих в нем расположиться.

Вы можете выбрать своему цыпленку лапки, ножки, крылышки, грудку по вкусу, можете оснастить его хвостовым, оперением, подвесить дополнительные топливные баки,

ракеты "воздух-воздух" и "воздух-земля", установить приборы ночного бомбометания и многое, многое другое.

Дальше программа может развиваться как STRATEGY/ACTION, а можно сразу перейти в кулинарный модуль с помощью описанного выше блока питания. Но в последнем случае способ предпусковой подготовки компьютера путем замачивания будет несколько отличаться, впрочем вместо инструкции к программе прилагается книга о вкусной и здоровой пище.

Проблем с ошибками больше нет!

Известно сколько неприятностей доставляют каждому работающему с компьютером сообщения об ошибках.

Наконец-то эта проблема решена всемирно известной фирмой PRESENT FROM AFRICA.

Покупайте новую программу "O.K. Simulator" (Имитатор O.K.)

Что бы Вы ни делали с компьютером, он всегда выдает радостное сообщение O.K.

Программа проста в работе. С ней успешно (и без ошибок) работают грудные дети. Группа академиков освоила работу с этой программой всего за полтора месяца во время учебного круиза по маршруту Одесса-Афины-Кейптаун-Сингапур-Токио и обратно. Теперь у них тоже все O.K.

По окончании круиза все академики высказали два пожелания и поставили один проблемный вопрос.

Первое пожелание - сократить количество утомительных нажатий клавиш хотя бы до двух (большее количество мешает круизу).

Второе пожелание - повторить круиз для окончательного решения вопроса о рекомендации программы ко всеобщему внедрению.

Проблемный вопрос: почему во всех городах мы побывали по два раза, а в Токио только один?

Сейчас фирма работает над новой версией, печатающей сообщение "O.K." в любых необходимых количествах автоматически, но, как сообщают, работа над проектом может затянуться из-за неразрешимости проблемного вопроса. Он стал известен в академических кругах под названием "Токийский синдром".

Ряд крупных академических институтов уже взялись за его отработку, по крайней мере на моделях. В модельных экспериментах функции Токио выполняли Лондон, Париж и Нью-Йорк, но проблема еще далека от разрешения. На очереди Сан-Франциско и Рио-де-Жанейро.

Происшествия!

В московском отделении фирмы PRESENT FROM AFRICA совершено крупное хищение. Украдены персональный компьютер, монитор, курсор и джойстик.

При обыске у подозреваемого обнаружены компьютер, джойстик и монитор.

Фирма гарантирует всем гражданам, которым известно что-либо о местоположении курсора, призовое вознаграждение своим программным обеспечением.

Без курсора фирма не может продолжать работы над замечательными проектами.

СОВЕТЫ ЭКСПЕРТОВ

PROFESSIOHAL TENNIS

"Dinamic" 1990 г.

Эксперт М.Аксюта г. Днепропетровск.

Эта игра относится к имитаторам спортивных игр. Вам предоставляется возможность хорошо потренироваться и сыграть на Уимблдонском турнире со звездами мирового тенниса. Но начнем по порядку. После загрузки игры перед

Вами основное меню:

0. TORNEO (турнир)
1. CONTROLES (органы управления)
2. EQUIPAMIENTO (экипировка)
3. ENRENAMIENTO (тренировка)
4. CARACTERISTICAS (условия игры)

1. CONTROLES

В этом режиме Вы можете ввести раскладку клавиатуры для управления двумя игроками:

TECLADO_A и TECLADO_B

2. EQUIPAMIENTO

Здесь Вы можете выбрать себе экипировку, а именно: - сначала кроссовки, а затем ракетку.

3. ENTRENAMIENTO

Вам предлагается потренироваться перед игрой на Уимблдоне. Вы можете выбрать тренировочный режим из следующего субменю:

1. PARTIDO
2. SAQVE
3. VOLEA

Выход из режима тренировки - нажатием клавиши "R".

Saqve - отработка подачи. Выбираете себе управляющий орган (например Кемпстон-джойстик) и проводите тренировку.

Нажмите кнопку "огонь" и переместите появившуюся на экране мишень в то место площадки, куда бы Вы хотели послать мяч. Подавать мяч нужно обязательно в накрест лежащий квадрат поля противника. Так, если Вы находитесь в правом нижнем квадрате, то подавать можно только в левый верхний. При ошибке или попадании мяча вне разрешенной зоны загорается сообщение и Вам дается вторая попытка, как в настоящей игре. Отработав подачу, можно нажатием "R" вернуться в субменю и перейти к отработке приема мяча.

Volea - отработка приема мяча. Ваш игрок может свободно перемещаться по площадке. Удар он выполняет нажатием кнопки "огонь". Если одновременно нажать "огонь" и "вверх", то удар получается более сильным, но пользоваться этим следует с осторожностью - когда Вы находитесь на задней линий или противник вышел близко к сетке, иначе мяч после Вашего удара может улететь за пределы площадки.

Если совместить удар с движением влево или вправо, то можно соответственно послать мяч влево или вправо, но это происходит не всегда и зависит от того, как Вы стоите относительно мяча. Так, если Вы стоите слева от мяча, он может полететь либо влево, либо прямо и наоборот, если Вы стоите справа от мяча во время приема.

Потренировавшись в приеме подачи и закончив тренировку клавишей "R". Вы можете

сыграть пробную игру с компьютером, нажав клавишу "1" - Partido и выбрав для него силу игры "ORDENADOR".

Если Вас при этом не устраивают кроссовки или ракетка, вернитесь в главное меню и поменяйте их.

4. CARACTERISTICAS

Вам предлагается следующее меню:

1. TIPO DE PISTA….
2. CAMBIO DE CAMPO….
3. NUMERO DE SET….

Tipo de pista - выбор покрытия корта, на котором Вы будете тренироваться. Вам будут предложены следующие варианты:

- TIERRA - грунтовое,
- HIERBA - травяное,
- RAPIDA - пластиковое.

Cambio de campo - определяет будет ли производиться смена сторон поля по ходу игры.

Если Вы хотите играть все время с одной стороны, выбирайте NO (НЕТ), если же желаете проводить смены полей, выберите SI (ДА). Начинающим рекомендуется выбирать NO.

Numero de set - количество сетов (1-3-5), которое будет разыграно с компьютером при тренировке.

0. TORNEO

Здесь разыгрывается турнирная игра. Сначала Вам предложат выбрать количество участников: 1-4 и ввести их имена, после чего Вы попадете в следующее субменю:

0 - CLASSIFICATION ATP
1 - INICIAR TEMPORADA
2 - CONTINUAR TORNEO
3 - TABLA DEL TORNEO

Classification ATP – таблица участников турнира, их номера. Вы всегда начинаете с самой низшей ступени.

Iniciar temporada - ввод имени игрока.

Continuar torneo - выбор управляющего органа и начало (продолжение) игры. Игра может быть прервана для перенастройки нажатием клавиши "R".

Tabla del torneo - таблица пар участников.

Итак, предварительно потренировавшись и выиграв у компьютера 3-4 сета, Вы можете принять участие в настоящем Уимблдонском турнире. Разве Вам не хочется победить Беккера или Лэндла?!

SNOOKER

Эксперт Ескевич А.А. г. Новосибирск.

"Снукер" - разновидность биллиарда.

Ныне эта игра - одна из популярнейших не только на туманном Альбионе, но и в более, чем тридцати других государствах. В Великобритании действует огромная сеть клубов, объединяющая более чем 6 млн. любителей снукера - это при том, что все население страны - 54 миллиона человек!

На биллиарде британцы играют уже свыше 100 лет. Таков же возраст и снукера. Происхождение этой игры точно не установлено. По одним сведениям снукер был придуман в Индии, а затем завезен в Англию, где и получил признание и широкое распространение. Другие знатоки утверждает, что в 1875 году желание как-то разнообразить обычный биллиард толкнуло одного из младших офицеров британской армии на то, чтобы включить в

игру шары другого цвета.

Что же касается самого слова "снукер", то оно произошло от английского "snook", что в переводе означает "длинный нос". Снукерами же в британской армии называли юных кадетов только что начавших службу.

Так что же такое снукер?

Точный удар, уверенная срезка - и по зеленому сукну с мягким стуком раскатилось 22 разноцветных шара, останавливаясь в неожиданных положениях…

Это снукер - почти неизвестная в нашей стране игра на биллиарде. Изобретенная во второй половине прошлого века, игра оказалась сложнее, чем известные "Американка" и "Русская пирамида". Она скорее пробуждает интерес к сложным движениям шаров, чем к выигрышу.

Правила игры.

В снукер обычно играют вдвоем, но могут играть и несколько игроков.

Принцип очередности удара простой: если удар не принес очков, бьет следующий игрок.

Цветные шары оцениваются в зависимости от цвета:

- 15 красных шаров - по 1 очку;
- один желтый - 2 очка;
- один зеленый - 3 очка;
- один коричневый - 4;
- один синий - 5;
- один розовый - 6;
- один черный - 7 очков.

Белый шар - это биток. Только им можно бить по остальным шарам.

Разбивающий пирамиду может установить биток в любом месте зоны "дома" (площадь, ограниченная полукругом) - там, откуда удобнее бить.

Исходная расстановка шаров показана на рисунке. Вместо цвета шагов мы поставили их оценку в очках.

Первый ударом нужно сыграть только красный шар, а если биток заденет за какой-нибудь другой, игроку засчитывается ошибка и списываются очки в зависимости от ценности затронутого шара. Самый первым ударом важно не только удачно разбить пирамиду, но и отогнать биток как можно дальше от нее, создавая другому игроку позицию посложней.

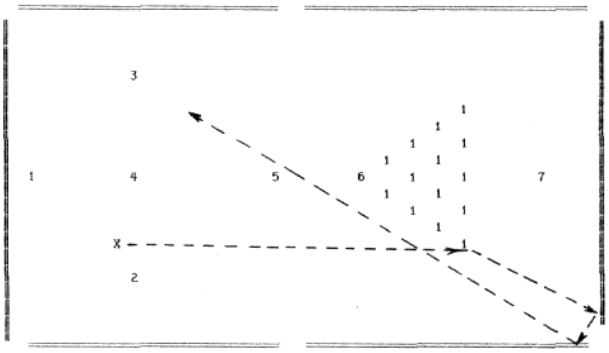
На рисунке классический начальный удар показан пунктиром.

Шары кладутся в любую из шести луз. За уложенный красный шар засчитывается одно очко. После этого биллиардист имеет право удара по любому цветному шару (цветными называются все шары, кроме красных и битка). Если начинающий игрок более уверен в каком-нибудь шаре, то он может бить и красный - один, другой, а потом, подогнав биток к цветному, уложить его в лузу.

Нужно твердо запомнить, что перед каждым цветным шаром должен быть забит красный!

Профессионалы снукера иногда усложняют игру тем, что после красного шара можно играть только цветной шар. Такая тактика всегда помогает набирать крупные серии. (Серия - сумма выигранных очков в течение одной очереди).

Например, вот какая серия может получиться, если было уложено в лузу несколько шаров: красный (1 очко), зеленый (3), снова красный (1 очко), розовый (6), красный (1), синий (5) и т.д.



Положенный в лузу красный шар выходит из игры. До тех пор, пока на столе есть хотя бы один красный шар, можно забивать цветные шары в любом порядке по своему выбору, но всякий раз перед цветным игрок должен уложить в лузу красный шар. Забитые цветные шары на этой стадии игры выставляются на те точки стола, где они стояли в начале игры.

Первая часть игры кончается, когда все красные шары забиты в лузы. Теперь правила по отношению к цветным шарам становятся более строгими: их надо класть в лузу в порядке их ценности, начиная с желтого (забив желтый, можно класть зеленый, затем коричневый, синий, розовый и черный. Уложенные таким образом цветные шары в игре уже больше не участвуют. Однако, если был сделан неправильный удар - биллиардист совершил ошибку при ударе, - цветной шар выставляется на свою основную позицию.

Игра подходит к завершению, когда на столе остается лишь 2 шара: "последний черный" и биток.

Бывает, что у партнеров примерно равное количество очков. Поэтому все решают эти семь очков за черный шар. Когда "последний черный" сыгран, или совершена ошибка - игра заканчивается. Редко, но может случиться и ничья - в этой случае для выявления победителя разыгрывается повторение "последнего черного". Черный шар выставляется на свое место, и по нему играют битком из "дома".

Когда знатоки биллиарда делают красивую игру, они не только укладывают шары в лузу, но при этом еще и маневрируют битком по всему полю. В результате маневра противнику может быть поставлен снукер - это кульминация игры, ее высший смысл, который и дал ей такое название. Снукер - это ловушка, из которой может выбраться только осторожный и хладнокровный игрок.

Снукер - это позиция, из которой биток не может достать тот шар, который по правилен должен быть сыгран: он либо "замазан", либо даже полностью закрыт, но шанс у Вас все-таки есть - выручит меткий бортовой удар.

А теперь ошибки, часто встречающиеся в снукере.

Случается, что игрок промахивается по шару, но это еще полбеды: этот шар, оттолкнувшись от борта, может задеть не тот, который должен быть сыгран. Если биток завалится в лузу, даже вслед за забитым шаром, эта ошибка наказывается штрафом в 4 очка. При назначении штрафа берется во внимание оценка шара, который следовало сыграть, или оценка шара, который сыгран неправильно. По правилам всегда берется наивысшее количество очков. Поэтому, если был правильно сыгран красный (или желтый, зеленый, коричневый) шар, но биток закатился в лузу, то списывается минимальное количество очков - 4; если же в таком положении окажется черный шар, штраф составит уже 7 очков (соответственно 6 - за розовый и 5 -за синий).

Когда нужно играть, например коричневый шар, и игрок делает правильный удар по нему, а закатывается какой-нибудь другой, то очки не списываются, а уложенный шар выставляется по правилам.

Подсчитано, что теоретический предел возможностей игрока в снукер - набрать за одну серию 155 очков (напомним, что серия - это сумма очков, выигранных за одну очередь).

Выдающийся мастер снукера англичанин Джо Дейвис установил в 1955 году рекорд мира, набрав серию в 147 очков. Может быть, Вы попробуете превзойти мастера?

Настройка программы.

После загрузки программа выдает следующее меню:

CONTROL OPTIONS	ОРГАН УПРАВЛЕНИЯ
1. KEYBOARD	1. КЛАВИАТУРА (O,P,Q,A,Enter)
2. KEMPSTON	2. КЕМПСТОН- ДЖОЙСТИК
3. CURSOR KEYS	3. КУРСОР (5,8,6,7,0)
4.INTERFACE 2	4. ИНТЕРФЕЙС 2 (Синклер-джойстик)
Press 5 to Continue	Нажмите "5" для продолжения.

Выбрав орган управления, нажмите "5" - появится следующее меню:

Game options:	Опции игры:
1. ONE PLAYER	- один игрок
2.TWO PLAYER	- два игрока
3. CURSOR SOUND ON	- звук курсора включен
4. CURSOR SOUND OFF	- звук курсора выключен
5. LONG GAME	- длинная игра
6. SHORT GAME	- короткая игра
7.CURRAH SPEECH OFF	- речь выключена
8. CURRAH SPEECH ON	- речь включена
Press any other key to play	Для начала игры нажните любую другую клавишу.

Сделав выбор, нажмите клавишу, и на экране появится биллиардный стол с расставленными шарами.

Структура экрана.

В левом верхнем углу показаны набранные Вами очки (POINTS) и сумма штрафных очков за ошибки (FOULS), в правом верхнем углу - номер серии (VISITS).

Внизу показаны сила удара (POWER), биток для выбора направления вращения шара (SPIN) и то, какой шар нужно сыграть в данный момент (например, "RED WANTED" - "Требуется красный").

После того, как Вы забьете красный шар, появится надпись "COLOUR WANTED" ("требуется цветной"). Выберите цветной шар, который Вы хотите забить и нажмите клавишу, соответствующую стоимости этого шара (2-желтый, 3-зеленый, 4-коричневый, 5-синий, 6-розовый, 7-черный). Например, Вы решили сыграть черный шар - нажмите "7" и надпись "COLOUR WANTED" сменится на "BLACK WANTED". После этого Вы обязательно должны бить по черному шару. Если промахнетесь, или биток заденет сначала за какой-либо другой шар, Вам начислят штрафные очки за удар не но правилам.

Начало игры.

Перед началом игры Вы должны установить биток в любой точке "дома". Для этого переместите курсор в выбранное место и нажмите "огонь" (эту операцию Вам придется проделывать всякий раз после того, как биток угодит в лузу).

Теперь Вы должны нанести удар. Это несложно, но требует тщательной подготовки:

1. Установите курсор в то место, куда хотите направить биток и нажмите "огонь".

2. Установите силу удара клавишами влево/вправо, пользуясь шкалой внизу экрана, и вновь нажмите "огонь".

3. Установите курсор на ту точку битка, куда Вы хотите "ударить кием" (проще всего бить в центр битка, но если Вы научитесь "срезать" шар, это будет весьма полезно) и вновь нажмите "огонь" - удар произведен.

Теперь, когда правила известны, остается лишь приобрести навык - и победа будет за Вами! Дерзайте, будущие мастера международной игры на биллиарде!

QUAZATRON

Сегодня мы представим Вам программу QUAZATRON, разработанную Стивом Тернером, с работами которого наши читатели уже знакомы по циклу статей "Профессиональный подход".

В мае 1986 года известный журнал "Sinclair User" присвоил этой программе свой знак:

Sinclair User
CLASSICS

Этот красный с золотом знак присваивается только тем программам, которые не просто являются лучшими в своем жанре, а открывают новые жанры или новые направления в своем жанре или открывают новые, ранее неизвестные приемы и методы. Для программиста получение такого знака может быть и не столь значительно, как получение премии "Оскар" для кинорежиссера, но не менее дорого.

Конечно, сейчас уже далеко не 1986 год, но игра эта еще во многом свежа. Возможно, что в силу определенной сложности, она долгое время выпадала из поля зрения наших пользователей и об этом приходится только сожалеть. Ведь программа настолько многопланова и разнообразна, что возвращаться к ней можно много и много раз. Она представляет довольно гармоничный баланс между играми аркадного и стратегического жанра.

Квазатрон - это огромный подземный технополис, населенный роботами. Расположен он на планете Квартех. Несколько попыток земных экспедиций по проникновению в этот комплекс закончились неудачно. Вам предлагается попробовать свои силы и захватить город с помощью специального робота MECHNOTECH KLP2.

KLP 2 был сконструирован в качестве разведывательного робота и провел специальную подготовку для участия в исследовательских экспедициях в составе разведывательных команд. К сожалению, его пришлось отстранить от групповых операций в связи с его патологической страстью разбирать другие устройства (обратите внимание на то, что KLP 2 читается по-английски как KLPTWO - очень похоже на КЛЕПТО - и вспомните, что непреодолимое желание тащить к себе все, что плохо лежит, называется клептоманией).

Отправленный для рутинной патрульно-таможенной службы на границы галактики, КЛЕПТО, тем не менее, завоевал известность и стая общественным героем, когда утилизировал левую ногу знаменитого межзвездного пирата, робота-андроида человекоподобного типа.

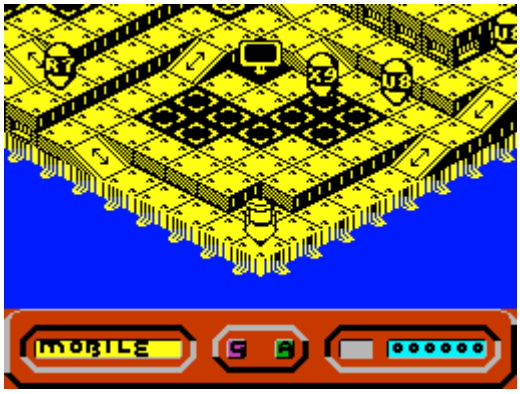
Вот почему о нем вспомнили, когда готовили миссию QUAZATRON.

Находясь под Вашим управлением, KLP 2 может работать в двух основных режимах:

- MOBILE - режим перемещения;
- GRAPPLE - режим захвата.

Когда он работает в режиме перемещения, игра развивается как обычная аркадная игра. Вы можете перемещаться по трехмерному уровню технополиса, можете периодически отстреливать враждебных роботов, можете перемещаться с уровня на уровень, можете подключаться к установленным в разных местах дисплеям или подпитываться энергией в специальных точках.

Подключение к дисплею позволяет войти в информационную сеть технополиса, через которую открывается доступ к самой разнообразной информации. Вы можете увидеть план уровня, схему расположения уровней в городе, можете получить справку о роботах-противниках, узнать их спецификацию.



Подзарядка энергией - тоже совершенно необходимое действие.

Как бы Вы ни работали, Ваш успех зависит от наличия энергии. Если ее мало, загорается надпись POWER, раздается неприятный звуковой сигнал и, если Вы сломя голову не броситесь к источнику энергии, можете считать, что миссия закончена. Самое интересное - это то, что для указания уровня энергии нет никаких счетчиков, никаких шкал, никаких указателей. Над головой KLP-2 есть небольшая "шапочка". Скорость, с которой она вращается, и зависит от уровня энергии.

Надо сказать, что обычное аркадное путешествие по этажам города закончится очень быстро. Дело в том, что среди населяющих его роботов есть такие, которые не дадут Вам ни малейшего шанса остаться в живых, если Вы попробуете перебежать им дорогу.

Вот тут-то аркадная игра и превращается в стратегическую (а если хотите, то в деловую, ведь менеджмент - тоже стратегия, хоть и отличается от военной). В этой фазе игры Вы работаете со своим роботом в режиме GRAPPLE. В этом режиме Вы можете войти в непосредственный контакт с противником, подключиться к его внутренним цепям и попытаться его перепрограммировать, если он не сделает то же с Вами. Здесь Ваш успех будет зависеть от быстроты реакции, от оснащенности Вашего робота и конечно от его энергетических возможностей.

Схватка в режиме GRAPPLE представляет собой аркадную вставку в основную игру и развивается на особом экране, но мы о ней писать не будем, предоставив чуть ниже слово нашему эксперту Курбацкому Андрею Александровичу из Казани, накопившему определенный опыт в этом деле.

Если Вам удастся захватить хотя бы семь блоков из двенадцати в логических цепях Вашего противника, то схватка считается закончившейся в Вашу пользу, робот противника переходит в подчиненное состояние и Вы можете приступить к его тестированию. Тестирование покажет, какие модули остались неповрежденными в результате схватки и Вы сможете принять решение о том, что Вам можно использовать для себя. Так, постепенно каннибализируя павших противников, Вы можете улучшать оснащенность своего робота.

Вот где открывается поле для стратегии.

Во-первых, Вы должны сами принять решение о том, с каким роботом Вам имеет смысл входить в тесный контакт, а какого достаточно просто расстрелять. Решение Вы принимаете, зная модель робота по его бортовому номеру, его спецификацию и состав его модулей.

Противник не может быть намного мощнее Вас - это шаг к могиле, но и не должен быть слишком слабым, т.к. если с него нечего взять, то и нет смысла расходовать на него время и силы, разве что для тренировки.

Во-вторых, сама схватка должна проходить с Вашей стороны достаточно интеллектуально, ибо после иного боя Вы можете уже не найти у противника необходимых Вам компонентов в исправном состоянии. Попробуйте в начале игры перестрелять мощных боевых роботов противника и Вам впоследствии уже негде будет взять особо ценные модули, необходимые для захвата всего города.

В-третьих, утилизация для своих целей исправных блоков от захваченных роботов тоже должна проходить с умом. Возьмите очень мощный двигатель - и с Вашим скромным энергетическим отсеком Вы устанете бегать на подзарядку, а если один раз не добежите - игре конец. Возьмите самое изощренное орудие и система защиты и на Вашем слабеньком

шасси KLP-2 будет еле ползать по поверхности, а падение с небольшой высоты станет приводить к значительным потерям энергии. И так далее и тому подобное.

Таким образом, игра представляет такой сплав стратегических решений, что ее серьезный анализ вполне может занять Вас не на одну неделю и доставит огромное удовольствие.

В заключение этого общего обзора программы прежде, чем предоставить слово нашему эксперту из Казани, мы приведем небольшое интервью со Стивом Тернером, и заглянем краем глаза на ту кухню, где готовятся такие вкусные блюда для Вашего компьютера.

В: Скажите пожалуйста, как долго шла работа над программой QUAZATRON?

О: Эта работа затянулась более, чем на шесть месяцев. Я знаю, конечно, что многие заявляют о том, что вполне пристойная аркадная игра может быть сделана и за два месяца, но у меня особый случай - ведь и музыку и графику я тоже делаю сам. Я вообще все делаю сам, хотя концепцию программы и наброски широко обсуждаю с коллегами.

В: Вам удается обыграть собственную программу?

О: Вы знаете, после полугода работы с машинным кодом, я месяца два вообще не мог смотреть на эту программу. Потом постепенно успокоился и нередко играю в нее. Но ответить на Ваш вопрос я не могу, ведь я не играю на выигрыш. У меня совсем другая цель - я ищу способы, которыми можно было бы "завалить" программу.

В: Когда Вы поняли, что эта программа Вам удалась?

О: Только в самый последний момент. Здесь надо знать мой метод работы над программами. Я не вижу ее до конца работы, все процедуры отрабатываются по отдельности или группами, но окончательная сборка производится в самый последний момент. Конечно, многие вопросы, особенно связанные с графикой, приходится обкатывать на моделях, ведь здесь важно быстродействие процедур с одной стороны и художественное впечатление с другой, но это совсем не то же, что работа с программой.

В: Что Вам не удалось в программе?

О: Отсутствие гладкого скроллинга экрана. У меня, конечно, были идеи как его обеспечить, но неминуемо приходилось бы жертвовать чем-то другим и сейчас, оглядываясь назад, я полагаю, что так получилось даже лучше.

В: Ваша программа чем-то напоминает PARADROID, выпущенный незадолго до нее для "Коммодора-64"? Она была в какой-то степени для Вас источником вдохновения?

О: Нет, все гораздо проще. Просто мы с Эндрю Брейбруком, автором "Парадроида", работаем в одной комнате. Естественно, по ходу работы идет постоянный обмен идеями. Кроме того, очень часто совместно обсуждаются вопросы дизайна. А если уж говорить об источнике вдохновения, то им явилась в некоторой степени программа MARBLE MADNESS и, в какой-то степени, ALIEN-8.

В: И последний вопрос. В самом начале программы звучит удивительно сложная музыка, если вспомнить, что у "Спектрума" только один канал...?

О: Да, конечно, каждый знает, что звуковая система "Спектрума" совершенно не соответствует прочим его возможностям. Но мне удалось программным путем, основываясь на системе прерываний, подготовить пакет процедур, имитирующих многоканальный звук. В конце концов, все это свелось только к манипуляциям фазой и частотой.

* * *

Теперь мы рассмотрим практические вопросы, связанные с боевыми действиями на планете Квартех. Слово имеет Курбацкий А.А., г.Казань.

Все роботы, с которыми Вы можете встретиться в технополисе, имеют бортовую маркировку, с помощью которой можно получить более подробную информацию об этом роботе.

Маркировка состоит из буквы и цифры. Буквенная маркировка указывает на класс робота, а цифровая - на положение робота в этом классе.

Классы роботов:

A (AUTOMATION)

B (BATTLE)
C (COMMAND)
L (LOGIC)
O (MEDIC)
R (REPAIR)
S (SECURITY)
U (UTILITY)
X (MENIAL)

Не входят в эту классификацию только два типа роботов. Это AB (ANDREWOID) И ST (PROGRAMMER).

Цифровая кодировка (от 0 до 9) указывает на статус робота в своем классе. Чем она ниже, тем выше статус робота, тем более совершенными системами он вооружен, тем больше доступный для него запас энергии.

Все роботы классов AB, ST, A и S относятся к высшей касте и защищены наиболее сильно. К средней касте можно отнести роботов L, O, R. К низшей касте относятся роботы U и X и защищены они довольно-таки слабо и в цифровой нумерации не опускаются ниже 7.

Нажав клавишу "огонь", Вы переходите в режим GRAPPLE и, коснувшись любого робота, входите в бой. Бой за перехват управления выглядит так (см. рис. 1).

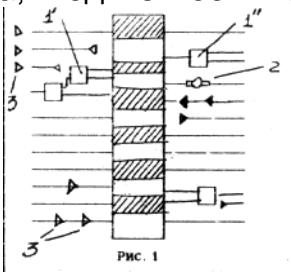


Рис. 1

В начале игры у Вас всего три энергетические капсулы (3) которые Вы можете подсоединить к любому проводу, соединенному центральным блоком (см. рис.). Иногда на проводах стоят такие устройства как сумматоры (делители) (1), инверторы (2) и дополнительные энергетические капсулы (3). С помощью делителя (1') можно от одного провода запитать два и более отсека, сумматор же (1'') наоборот от несколько проводов запитывает один отсек. Инвертор (2) обращает Ваш сигнал и превращает в сигнал противника, так что Вам он мешает, а не помогает. Дополнительные энергетические капсулы могут помочь Вам увеличить время подпитки отсека, так как при израсходовании энергии в Вашей капсуле, подпитка блока прекращается.

Некоторые провода могут быть оборваны. Сбоку указано количество энергетических капсул, имеющихся в Вашем распоряжении.

Сначала Вам предоставляется право выбрать сторону, с какой Вы хотите вести бой. Это очень важный момент, от которого в значительной степени будет зависеть успех боя. После выбора стороны Вы можете с помощью клавиш "вверх", "вниз" подвести энергетическую капсулу к проводу, к которому вы хотите подключиться, и нажать клавишу "огонь". Блок центрального отсека закрашивается. Конечная цель боя - захватить отсеков больше, чем Ваш противник. Всего на бой дается 60 секунд, желательно подключиться чуть позже, чем противник, так как это позволит при Вашем недостатке энергетических капсул продержаться как можно дольше и, в конечном итоге, захватить энергетический отсек. Если схватка закончится вничью, то Вам предоставляется другая энергетическая схема.

В случае победы противника, если это первый бой, Ваш робот разрушится, а если эта схватка происходила после того, как Вы уже кого-либо победили, то теряются все захваченные ранее блоки и оружие, а силовая установка остается почти без энергии. Тогда Вам после схватки нужно быстрее идти к энергетической клемме, расположенной на этаже, и подпитаться энергией. Маркировка последнего побежденного робота отображается в окне чуть правее центра пульта. Вот список устройств, которые Вы можете захватить, расположенный по степени приоритета.

Двигательная установка (DRIVE).
LINEAR MK 1
LINEAR ME 2
LINEAR MK 3
GRAVITRONIC MK 1
GRAVITRONIC MK 2
HEAVY DUTY
DUAL LINEAR
ULTRAGRAV

Оружие (WEAPONS)
PULSE LASER
DUAL LASER DESINTEGRATOR
AUTOCANNON
DISRUPTOR

Наибольшей разрушительной способностью обладает DISRUPTOR, который поражает всех роботов, видимых на экране. AUTOCANNON стоит на втором месте: пробивает любой корпус, но стреляет только в одном определенном направлении.

Имеет значительный вес. Все прочее оружие пробивает лишь определенные корпусы, да и то не всегда с первого раза.

Силовые установки (POWERS)
CHEMIFAX MK 1
ROBOTRONIC MK 1
ROBOTRONIC MK 2
ROBOTBONIC MK 3
TRIOBATIC
CYBONIC MK 1
CYBONIC MK 2

Чем выше класс силовой установки, тем более мощный двигатель и оружие в течение более долгого времени способна она поддерживать.

Шасси (CHASSIS)
DURALITE
TRIALIUM MK 1
TRIALIUM MK 2
CHROMITE
PLASTEEL
STRESSED PLASTEEL
CORALLOID MK 1
CORALLOID MK 2

Чем совершеннее шасси, тем легче Ваш робот перенесет падение с высоты.

Другие устройства (DEVICES).

Эти устройства являются дополнительными и потому они есть только у роботов высшего и среднего классов, у низших роботов они отсутствуют.

DISRUPTOR SHIELD - защита от воздействия DISRUPTOR-а. Она позволяет продержаться более долгое время при действии мощного оружия. Кстати, роботов, имеющих эту защиту, легче победить в борьбе за перехват управления, чем уничтожить оружием.

LASER SHIELD - защита от действия лазерного оружия, кстати неплохо защищает и от попыток пробить Ваш робот с помощью лобового тарана.

RAM THRUSTER - ускоритель. Позволяет увеличить скорость движения Вашего

робота, способствует успеху тарана.

POWER BOOTS - обеспечивает уменьшение нагрузки на шасси за счет дополнительной тяги, что позволяет более стойко переносить падения с высоты.

DETECTOR - устройство, предназначенное для контроля за уровнем радиации. Сигнализирует о наличии опасного излучения.

OVERDRIVE - устройство для форсирования тяги двигательной установки.

Для тех, кто интересуется тем, какие устройства установлены на роботах, приводим список в таблице 1. Список не полный в связи с тем, что пока еще не все обнаружено. Желаем удачи в игре.

обозначение	класс	двигатель	силовая установка	оружие	шасси	дополнительное устройство
AB	альфа	ULTRAGRAV	CYBONIC mk 2	AUTOCANNON	CORRLOID mk 2	RAM TRUSTER
ST	бета	DUAL LINEAR	CYBONIC mk 1	DESINTEGRATOR	CORRLOID mk 1	DISRUPTOR SHIELD
		AUTOMATION				
A1	альфа	ULTRAGRAV	CYBONIC mk 2	AUTOCANNON	CORRLOID mk 2	LASER SHIELD
		BATTLE				
B2	бета	HEAVY DUTY	TRIOBATIC	DISRUPTOR	CORRLOID mk 1	DISRUPTOR SHIELD
B3	гамма	GRAVITRONIC mk 2	TRIOBATIC	DISRUPTOR	STR.PLASTEEL	DISRUPTOR SHIELD
B4	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 3	DISRUPTOR	STR.PLASTEEL	LASER SHIELD
B5	дельта	GRAVITRONIC mk 1	ROBOTRONIC mk 2	AUTOCANNON	STR.PLASTEEL	RAM TRUSTER
B6	дельта	GRAVITRONIC mk 1	ROBOTRONIC mk 2	DESINTEGRATOR	CORRLOID mk 2	LASER SHIELD
B7	дельта	GRAVITRONIC mk 1	ROBOTRONIC mk 2	DESINTEGRATOR	CORRLOID mk 1	RAM TRUSTER
		LOGIC				
L3	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 2	DUAL LASER	PLASTEEL	POWER BOOTS
L4	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 2	PULSE LASER	PLASTEEL	
L5	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 1	PULSE LASER	PLASTEEL	
L6	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 1	PULSE LASER	PLASTEEL	
		MEDIC				
00	гамма	GRAVITRONIC mk 2	ROBOTRONIC mk 1	PULSE LASER	PLASTEEL	POWER BOOTS
		REPAIR				
R5	дельта	LINEAR mk 3	CHEMIFAX mk 2	DUAL LASER	PLASTEEL	DISRUPTOR SHIELD
R6	дельта	LINEAR mk 3	CHEMIFAX mk 2	DUAL LASER	PLASTEEL	
R7	дельта	LINEAR mk 3	CHEMIFAX mk 2	DUAL LASER	TRIALIUM mk 2	
R8	дельта	LINEAR mk 3	CHEMIFAX mk 2	PULSE LASER	TRIALIUM mk 2	
		COMMAND				
C1	альфа	ULTRAGRAV	CYBONIC mk 1	DESINTEGRATOR	CORRALOID mk 1	POWER BOOTS
C2	альфа	DUAL LINEAR	CYBONIC mk 1	DESINTEGRATOR	CORRALOID mk 1	DETECTOR
C3	бета	DUAL LINEAR	CYBONIC mk 1	DESINTEGRATOR	CORRALOID mk 1	OVERDRIVE
C4	бета	DUAL LINEAR	CYBONIC mk 1	DESINTEGRATOR	STR.PLASTEEL	LASER SHIELD
C5	бета	HEAVY DUTY	TRIOBATIC	DESINTEGRATOR	STR.PLASTEEL	DISTRUPTOR
		SECURITY				
S2	бета	HEAVY DUTY	TRIOBATIC	AUTOCANNON	STR.PLASTEEL	DETECTOR
S3	бета	DUAL LINEAR	TRIOBATIC	AUTOCANNON	STR.PLASTEEL	DETECTOR
S4	бета	HEAVY DUTY	TRIOBATIC	DESINTEGRATOR	STR.PLASTEEL	DETECTOR
S5	бета	HEAVY DUTY	ROBOTRONIC mk 3	AUTOCANNON	STR.PLASTEEL	DETECTOR
S6	бета	GRAVITRONIC mk 2	TRIOBATIC	AUTOCANNON	STR.PLASTEEL	DETECTOR
		UTILITY				
U7	епсилон	LINEAR mk 2	CHEMIFAX mk 1	PULSE LASER	TRIALIUM mk 2	
U8	епсилон	LINEAR mk 1	CHEMIFAX mk 1	PULSE LASER	TRIALIUM mk 2	
U9	епсилон	LINEAR mk 1	CHEMIFAX mk 1		TRIALIUM mk 1	
		MENTAL				
X8	епсилон	GRAVITRONIC mk 2	CHEMIFAX mk 1		TRIALIUM mk 1	
X9	епсилон	LINEAR mk 1	CHEMIFAX mk 2		DURALITE	

CAPTAIN FIZZ

"Psyclapse" 1989 г.
Эксперт Троекуров В.И.
г. Киев

Программа "CAPTAIN FIZZ" принадлежит к аркадному жанру. Действие происходит на космическом корабле "ИКАРУС", который потерял управление и движется к Солнцу. Если он подойдет к нему слишком близко, то взорвется, разрывая галактику. Чтобы это предотвратить, необходимо телепортироваться на борт звездолета и разрушить компьютеры, управляющие его движением.

Палубы корабля представляют собой достаточно сложную систему лабиринтов и Ваша главная цель разобраться в этой системе, исследовать корабль и восстановить его управление.

Характерной особенностью игры является то, что она позволяет работать над главной задачей "кооперативно". Два игрока могут играть одновременно и помогать друг другу своими действиями.

Экран программы.

Экран разделен для двух игроков по горизонтали. Панель каждого игрока состоит из трех частей:

1. Информационное табло. Сюда выводится текущий результат игры (очки), шкала, показывающая температуру Вашего оружия (когда шкала исчезла от продолжительной стрельбы, необходимо отпустить кнопку и подождать, пока она восстановится).

ARMOUR - Ваша зашита.

DAMAGE - Величина повреждений.

CHARGE - Заряд.

CREDIT - Количество "попыток".

2. Вторая часть - основной экран, на котором происходит действие игры. У каждого из игроков свой отдельный экран, что способствует независимости одного игрока от другого. Сверху над экраном расположены два слова:

HEALTH - состояние Вашего здоровья в 9999 энергетических единицах.

CARDS - карточки (пропуска), количество пропусков, их цвета (для каждого пропуска своя дверь, которую он может открыть).

3. Третья часть - индикатор состояния. Ломаная линия на нем - Ваша кардиограмма. По мере ухудшения Вашего самочувствия линия выпрямляется.

Здесь же представлены три дополнительных индикатора.

а) EXIT - "выход". Когда Вы ликвидировали все компьютеры на определенном уровне, то увидите, что загорится этот индикатор - значит можно телепортироваться на другой уровень. Открытый выход имеет овальную форму черного цвета с желтой окантовкой по краям. Закрытый выход имеет ту же форму но сверху на нем решетка желтого цвета.

б) SWITCHES - "переключатели". Здесь расположены четыре индикатора. Когда

индикаторы горят, силовое поле отключено.

в) MINES - "мины". Индикатор предупредит Вас, что в помещении имеется мина и лучше здесь не стрелять, т.к. при попадании в мину вы потеряете при ее взрыве определенную часть энергии.

Программа имеет еще одну интересную особенность. После загрузки ее можно настроить на работу с одним из трех европейских языков:

1. Английский.
2. Немецкий.
3. Французский.

После ввода программы, нажмите клавишу "5" на клавиатуре (2 игрок) и "FIRE" на джойстике (1 игрок) и начинайте игру.

Итак, начинаем игру вдвоем. На экранах появятся два героя - один желтого цвета, другой - белого. Первый управляется от клавиатуры клавишами:

1 – влево 2 – вправо
3 – вниз 4 - вверх
5 - "огонь"

Второй герой может управляться как от джойстика, так и от клавиатуры:

6 – влево 7 – вправо
8 – вниз 9 - вверх
0 - "огонь"

Первое, что Вы должны сделать - это расстрелять компьютеры (они выглядят, как белые квадратики, маркированные буквой "L").

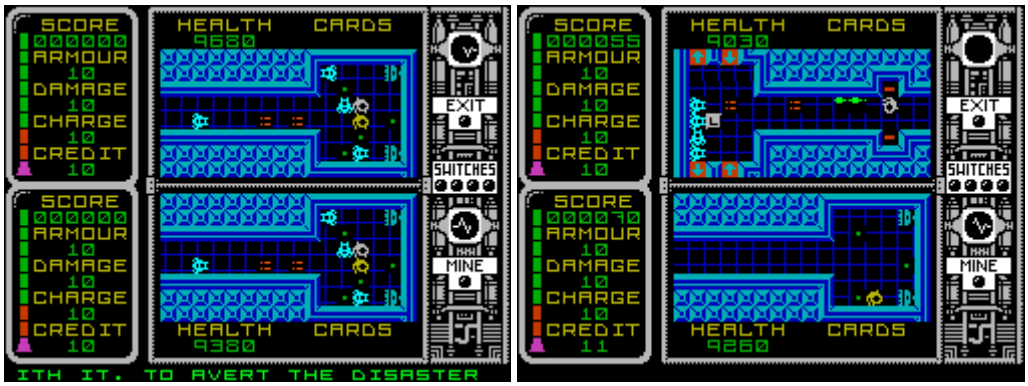
Сложность заключается в том, что их охраняют боевые роботы, с которыми предстоит провести перестрелку. Кроме того, есть небольшое силовое поле, способное какое-то время защищать компьютеры от поражения Вашим оружием.

Оба космонавта имеют по мощному бластеру с неограниченным боекомплектом. Но Вам следует внимательно следить за температурой своего оружия, так как оно не допускает перегрева и перестает стрелять. Кроме бластеров у космонавтов еще есть и супербомба, способная уничтожать враждебную боевую технику в пределах данного экрана, но не повреждающая компьютеры.

В исходном состоянии Ваша сила составляет 9999 энергетических единиц. Но здоровье можно поправить, когда Вы соберете энергетические аптечки (выглядят, как зеленые точки), но использовать их для своих целей можно только лишь подойдя к энергетической емкости. Чтобы понять, как выглядят энергетические емкости, надо обратить внимание на первую картинку игры. Рядом с двумя героями изображены зеленые точки (энергетические аптечки), а справа энергетические емкости.

Возьмите аптечку и подойдите к энергетической емкости. Вы увидите, что Ваша энергия возрастет до 9999 единиц - это предел и не тратьте лишние аптечки зря.

Карточки-пропуска понадобятся Вам, чтобы пройти в другие отсеки корабля. Пропуска выглядят как маленькие цветные прямоугольники. Они бывают 5 разных цветов: красного, желтого, синего, пурпурного и голубого. На дверях указан цвет пропуска, которым они открываются. Если Ваш пропуск не соответствует цвету на двери, то Вы не сможете ее открыть.



Есть двери, вырабатывающие защитное силовое поле, контактов с которым

необходимо избегать. В случае Вашей гибели задача партнера очень усложнится. Впрочем, если ему удастся пройти на следующий уровень, Вы появитесь снова, хотя уровень Вашей энергии при этом будет равен только 2000 единиц.

Гибель обоих игроков ведет к прекращению игры. После этого программа предложит Вам продолжить игру (CONTINUE). Утвердительный ответ (Y) позволяет вернуться на один-два уровня назад и продолжить игру. Отрицательный ответ (N) позволяет начать игру сначала.

Выключить силовое поле двери можно переключением четырех тумблеров. Их надо отыскать на данном уровне и определить правильную последовательность их включения. При правильном их положении под индикатором SWITCHES должны загореться четыре зеленых лампочки - поле отключено и можно идти. Тем не менее будьте осторожны - некоторые поля отключаются лишь на несколько секунд, что достаточно для входа, но для выхода придется переключать тумблеры еще раз.

Силовые поля, боевые роботы, мины - это еще не все неприятности, подстерегающие Вас. Наиболее серьезным препятствием является боевая механизированная установка, стреляющая самонаводящимися ракетами. Поразить ее без бомбы невозможно, но можно выстрелом из бластера изменить направление полета ракеты. Ракеты, не найдя нужной цели, самоликвидируются и Вы можете постараться как-то использовать это в своих целях.

На многих уровнях Вы встретите в красных квадратиках стрелки, показывайте определенное направление. Это транспортеры. Если Вы пойдете на стрелку, то пройдете сквозь нее, но назад здесь же вернуться не сможете, так что будьте внимательны.

Полезные советы.

1. Не играйте с партнером в перестрелку. Во-первых, это непроизводительные потери личной энергии, а во-вторых, при столкновении ваших снарядов происходит блокировка этого места и Вы не сможете его пройти.

2. Как пройти вдвоем через дверь, если пропуск есть только у одного героя?

Делайте так: Герой, у которого есть пропуск, проходит через эту дверь и поджидает пока партнер не окажется напротив двери. Затем он подходит к двери, она открывается и пропускает партнера.

3. Главное правило: работу в отсеках космонавты выполняют по одиночке, а имущество собирают поровну. Если в комнате находятся два пропуска, то Вам надо взять один, а второй на всякий случай оставить партнеру. Если же пропуск всего один, значит кто-то из Вас должен его использовать, в то время, как второй космонавт должен находиться снаружи и страховать напарника, чтобы быть готовым в любой момент прийти к нему на помощь.

FORUM

Нам кажется, что программа ELITE столь же неисчерпаема, как и сам "ZX-Spectrum". Вот и сегодня пришло письмо, которое, возможно, откроет новые перспективы для тех, кто уже несколько раз прошел программу от начала до конца и ищет, чем бы теперь заняться.

Наш читатель из Киева Руслан Хоминич провел исследование трех разных версии игры, нашел в них существенные отличия и, если Вы хорошо освоили одну из версий, Вам может быть интересно попробовать свои силы и в другой.

Версия 1.

Программа вскрыта командой "JOYSTICK CLUB". После загрузки появляется надпись "Press SPACE Commander" и изображается корабль "ASP MK II".

Это самая легкая версия игры. Ваш корабль довольно активно атакуется, независимо от Вашего правового статуса, но погибнуть довольно сложно.

В этой версии сравнительно неэффективны ракеты - они уничтожают не более половины заряда энергетического отсека и, наверное, поэтому многие пилоты их не любят, считая дорогим и малоэффективным оружием.

В этой версии можно даже брать на таран корабли противника, причем статус FUGITIVE Вы за это не получите, даже если протараните мирный корабль.

Не встречается корабль ADDER.

Очень велики расстояния между планетами, топлива едва хватает для перелетов внутри круга, а на периферии круга расстояние уже слишком велико для перелета в один ход. Это же мешает использовать в полной мере клавишу F для перехвата таргоидов в гиперпространство, т.к. после возврата обычно не хватает топлива на обратный путь.

Атака на станцию не дает видимых результатов - станция просто не принимает Вас без каких либо объяснения. Вблизи станции корабли не встречаются.

Эту версию стоит рекомендовать начинающим пилотам.

Меняйло Е.В. из Калуги уточняет, что в этой версии самое короткое расстояние между соседними планетами составляет 3,2 св. года, подтверждает отсутствие корабля ADDER и добавляет, что никак не проходит заправка топливом от звезды.

Версия 2.

Программа вскрыта командой "TENARK SOFTWARE".

Главное отличие - отсутствуют корабли KRAIT. На заставочной картинке представлен корабль ADDER. В связи с этими обстоятельствами, в программе изменилась тактика пиратов.

В целом программа представляет средний уровень сложности.

Как и в предыдущей версии здесь ракеты малоэффективны. При попытке атаковать станцию, Вам выдают угрожающую надпись и Вы лишаетесь возможности стыковки

станция оснащена системой ECM.

Эта версия позволяет стать миллионером в галактике 47 с помощью SAVE.

В обеих версиях имеют место захваты корабля пиратами при стыковке со станцией. Есть гипотеза, что Ваш корабль захватывается пиратами как бы "изнутри" в результате того, что Вы подбирали в космосе контейнеры с "рабами" и перегрузили свой корабль. Во всяком случае, не отмечалось захватов корабля, если в трюме не было "рабов".

Эта гипотеза имеет и то косвенное подтверждение, что были неоднократно отмечены захваты корабля в космическом пространстве "изнутри" после выхода из гиперперехода (об этом пишет, например, К. Дзреев из Ростова на Дону).

Версия 3

Версия имеет сообщение "M128".

Это наиболее сильная версия, она также наиболее соответствует фирменной инструкции.

Например, при попытке атаковать станцию, Вам навстречу вылетает стая полицейских кораблей и, хотя они не очень хорошо вооружены, их слишком много, чтобы устоять. Не спасает и перелет на другую планету, отныне в этой галактике Вы - изгнанник.

Значительно усложнена стыковка. Вокруг станции постоянно находятся различные корабли (преимущественно типа PYTHON) и, если Вы замешкаетесь, то рискуете с кем-нибудь столкнуться перед самой стыковкой. Часто это приводит к гибели.

Не менее опасно и резко тормозить после вылета со станции.

В этой версии большой разброс цен на некоторые запрещенные к перевозке товары, то есть Вас стимулируют к рискованным операциям. За Вами внимательно наблюдают. Если в глубинах космоса Вы совершите преступление, Вас могут не пустить потом на станцию. Здесь очень неприятно получать статус FUGITIVE. Не открывайте огонь первым, пока не увидите модель корабля визуально или пока он сам не начнет стрельбу.

Корабли типа KRAIT, THARGOID и SIDEWINDER здесь всегда пираты (возможно и ADDER, но пока он не встречался).

С кораблями типа PYTHON надо вести себя аккуратно. После первых выстрелов он выпускает KRAIT и SIDEWINDER - с ними можете воевать, это Ваша добыча, но головной корабль лучше оставить в покое и побыстрее покинуть поле боя.

FER-DE-LANCE - желательная добыча. Денег за него не дают, но рейтинг хорошо нарастает.

Нередко попадаются отшельники на астероидах. После обстрела они либо уходят, слабо маневрируя, либо отстреливаются, либо выпускают корабли KRAIT или SIDEWINDER. Из них можно "выжать" и контейнер с грузом, но поскольку отшельники защищены всегалактическим правой, Ваш статус может резко измениться.

В этой версии программы весьма эффективны ракеты. Попадание почти полностью выбивает защиту. Первым делом надо покупать систему ECM. Точно так же опасен и таран.

Сложность игры нарастает по мере повышения Вашего рейтинга. Сражения в галактике 1 намного проще, чем во второй галактике. Если в первой галактике Таргоиды поражались даже пульсирующим лазером, то во второй справиться с ними удалось далеко не сразу. Приходится иногда перестраивать тактику боя.

Фокус с 47-ой галактикой здесь не проходит.

Интересно, что отгрузочный блок имеет длину 104 байта. Видимо, это сделано для того, чтобы нельзя было использовать отгрузки с более легких версии.

Высокие качества этой версии подчеркивает также Сергей Дегтярев из Луганска. К сказанному выше мы можем добавить из его бортжурнала то, что при слишком высоком статусе FUGITIVE на станциях не открывается входное отверстие.

Кроме этих версии есть и другие, например версия, помеченная Jack O'Lantern. В ней стоит отметить отсутствие корабля KRAIT (сообщает Меняйло Е.В.). Условно назовем ее ВЕРСИЯ 4.

Лешинский А.М. (Н.Новгород) предполагает, что всего разных версии может быть порядка десятка, очень советует ввести какую-то систематизацию этих версий и упоминает о версии Родионова (назовем ее ВЕРСИЯ 5) и о версии Be-Be Soft (пароль 7Q) - назовем ее условно ВЕРСИЯ 6. К сожалению, он не приводит характерных особенностей данных версий.

"ИНФОРКОМ" горячо поддерживает предложение систематизировать версии и просит присылать свои предложения и идеи. В то же время, необходим какой-то универсальный параметр, который мог бы быть принят за основу классификации. Все читатели в письмах указывают на сообщения, сопровождающие или заканчивающие загрузку, но может быть целесообразно для каждой версии давать и раскладку по длинам входящих файлов?

* * *

У нас были вопросы от желающих получить дисковую версию программы ELITE - для них мы даем следующее сообщение:

Вышлю всем желающим дисковую версию ELITE ("Joystick Club") с записью состояния игры на диск.

169740, Коми, Усинский р-н, п. Приполярный, а/я 212. Сайфутдинов Е.В.

Сагдеев Р.Р. из г. Магнитогорска тоже работал с дисковой версией, вскрытой JOYSTICK CLUB, но в последнее время перешел на версию, в которой во время загрузки появляется надпись "M1 LOADING". Эта версия выглядит намного интереснее, отличается тем, что в ней есть очень сильные пираты и во много раз более серьезная полиция. Может быть, это что-то похожее на ВЕРСИЮ 3 по классификации Хоминича.

Перебросить ее на диск по блокам удается несложно с помощью копировщика PCOPIER, AMCOPY или каким-либо другим способом, загрузчик может быть таким:

```
10 BORDER NOT PI: PAPER NOT PI: CLS: CLEAR VAL "24751": RANDOMIZE USR VAL "15619": REM:
      LOAD "elite 1" CODE
20 RANDOMIZE USR VAL "15619": REM: LOAD "elite 2" CODE
30 RANDOMIZE USR VAL "24792": RANDOMIZE USR VAL "15619": REM: LOAD "elite 3" CODE VAL
      "16464"
40 REM POKE 46848,201
50 RANDOMIZE USR VAL "24795"
```

К сожалению, этого недостаточно, чтобы и отгрузка отложенного состояния игры и подгрузка тоже производились с дисковой поддержкой. Сагдееву Р.В. мешает это сделать отсутствие информации по управлению дисководом из машинного кода. Кстати, наш корреспондент отмечает, что адреса USR для старта блоков отличаются на 3. Это характерно для программ, сопровождающихся при загрузке надписью "M1 LOADING".

Перелеты к двойным звездам и невидимым звездам.

Мы уже писали о том, что в некоторых галактиках были обнаружены двойные звезды. Как уже было установлено, невидимые звезды - это тоже двойные звезды. К сожалению, не ко всем из них есть возможность слетать по желанию.

Меняйло Е.В. разработал схему перелета, основанную на безтопливном перелете со станции на станцию, о чем мы уже писали. (См. N11-12 "ZX-РЕВЮ-91", с. 253). В системе из пары звезд он называет одну основной, а вторую - подчиненной и предлагает следующий порядок действий.

1. Перелетите на основную планету любым способом.

2. Вызовите карту и с помощью поиска планеты по ее названию (клавиша "R") установите курсор на подчиненную планету. При помощи клавиши "P" это делать нельзя, т.к. информация имеется только об основной планете и при этом курсор будет всегда устанавливаться на основную планету.

3. Не нажимая клавишу "F" вылетите со станции и при помощи безтопливного перелета переместитесь на подчиненную планету. Оказавшись на ней, Вы можете просмотреть сводку цен, купить или продать товар, но при запросе информации Вам будут выданы данные только об основной планете.

Успешно посетил двойные звезды и наш читатель Владимир Кладов из Новосибирска. Он называет безтопливный перелет "перелетом-180", но пользоваться им не стал, а ввел в программу пару POKES.

POKE 60896,233

POKE 56272,233

Первый отключает поиск ближайшей планеты по команде "F", второй - по команде "H". Выдается информация (выполняется перелет) на планету, найденную предыдущей командой "O","R". Кстати, он приводит еще пару POKES.

POKE 56260,0 - "вечный межгалактический гипердвигатель".

POKE 28822,0 - "вечная энергетическая бомба".

Тайные возможности компьютера.

В эти апрельские дни, кажется решилась одна из важнейших проблем многочисленных любителей "Спектрума". Теперь каждый из Вас может, если захочет, в несколько минут конвертировать свой 48-ми килобайтный компьютер в полноценную 128-

килобайтную машину, причем сделает это абсолютно бесплатно.

Но, прежде чем мы перейдем к практике, несколько слов о сути.

Вы, конечно знаете, что Ваш Синклер-совместимый компьютер имеет 16 килобайтов памяти в ПЗУ и еще 48 килобайтов в ОЗУ, то есть всего 64K, а точнее 65535 байтов. Может он иметь больше? Наверное да, но ведь процессор Z-80 не всемогущ. Он работает с 16-разрядной адресной шиной и потому может обслуживать только 65535 байтов одновременно и не больше. Мы об этом писали в 1991 году на страницах "РЕВЮ", когда говорили о 128-килобайтных машинах. В них дополнительные 16-килобайтные блоки (называемые страницами) "впечатываются" на место страниц обычной памяти, подменяя их на время. Так что и в нем процессор работает всегда только с 64 килобайтами памяти.

А теперь рассмотрим такой вопрос. Вы, очевидно знаете, что Ваш компьютер в состоянии обслуживать еще и 65535 адресов внешних портов. Если Вы не работаете ни с какой периферией, то эта возможность просто никак не используется. А какая с точки зрения процессора разница - обслуживать внешние адреса или внутренние? Никакой. Вот если бы удалось так переключить процессор, чтобы он работал одновременно и с этими дополнительными адресами, как с оперативной памятью - Вы бы сразу имели 128 килобайтную машину без необходимости что-то паять и отлаживать.

Теперь мы Вас порадуем - такая возможность есть. Открыты два нигде не документированных регистра компьютера - раскрыта самая глубокая тайна К.Синклера, который уже в первых своих моделях предусмотрел их последующее расширение до 128K. Впрочем, нас интересует только первый регистр. Этот регистр называется первым Альтернативным Программным Регистром (сокращенно АПР.1). И за этим названием скрыта суть. Пожалуйста не путайте с альтернативным набором регистров процессора. Этот регистр к процессору не имеет никакого отношения и конфигурируется программным путем в результате серии сложнейших последовательностей команд IN и OUT.

Чтобы не забивать Вам голову, мы просто привели программу в машинных кодах, которая выполняет всю эту работу сама. К тому же мы еще сами не до конца разобрались, как же это все работает.

Наберите эту программу, запустите и наслаждайтесь грандиозным эффектом, до тех пор, пока компьютер не будет выключен. Вы можете чувствовать себя ничуть не хуже, чем любой владелец дорогой128-килобайтной машины, во всяком случае с этого момента жить Вам будет веселее.

Правда, после запуска этой программной последовательности, Вы не сможете получить того исходного меню, которое есть в стандартных 128-килобайтных "Спектрумах", но с этим ничего не поделаешь - ведь ПЗУ у вас осталось от 48-килобайтной машины. Хотя, если подойти к задаче творчески, мы уверены, что кому-то из Вас удастся развить идею и получить меню.

Первая часть программы представляет обычный БЕЙСИК-загрузчик, который загружает машинный код, начиная с адреса 32768 и запускает его с адреса 32786. При загрузке проверяется также правильность Вашего набора кодов, которые хранятся в строках DATA.

После запуска Вам придется немного подождать (секунду-другую), пока программа "прощупает" архитектуру Вашей машины и выполнит необходимые операции.

Желаем успеха!

```
10 CLEAR 32767
20 LET x=32768
30 FOR i=0 TO 14
40 LET c=0
50 FOR j=1 TO 6
60 READ a
70 POKE x,a
80 LET x=x+1
90 LET c=c+a
100 NEXT j
110 READ a
120 IF a<>c THEN PRINT "ERROR IN LINE "; 10*i+160: STOP
```

```
130 NEXT i
140 RANDOMIZE USR 32786
150 STOP
160 DATA 22,0,0,17,7,16,62
170 DATA 3,65,80,82,73,76,379
180 DATA 32,49,45,83,84,32,325
190 DATA 62,2,205,1,22,1,293
200 DATA 7,0,17,0,128,205,357
210 DATA 60,32,6,64,197,1,360
220 DATA 11,0,17,7,128,205,368
230 DATA 60,32,193,16,243,17,563[1]
240 DATA 3,7,205,84,31,210,540
250 DATA 123,27,118,118,20,203,609
260 DATA 154,28,203,155,213,122,875
270 DATA 205,155,34,230,248,179,1051
280 DATA 33,0,88,17,1,88,227
290 DATA 1,255,2,119,237,176,790
300 DATA 209,24,219,0,0,0,452
```

Советы и секреты.

В ответ на просьбу о пароле к 8-му уровню игры IMPACT пришло немало писем от читателей. Так, Сельдемишев М.М. из г. Томска называет этот пароль - J.D. но ставит новый вопрос: - время от времени в этой программе в левом верхнем углу появляется буква "B" - что бы это значило? Рожков П.В. из Пензы не только дает этот пароль, но еще сообщает POKE 54500,183. Этот POKE уже есть в загрузчике программы, но стоит после оператора REM и потому не действует. Достаточно стереть REM и запустить программу. С другой стороны, у него есть просьба к обладателям игры TANTALUS (на кассетах ходит под названием TANTAL.KEY) - не может ли кто-нибудь помочь с POKES - игра хорошая, но пройти ее не удается. С помощью нашего трехтомника по программированию в машинных кодах ему удалось самостоятельно отыскать POKES в ряде программ и он рад ими поделиться:

FROST BYTE	33052,0 – жизнь
	33805,52 – время
DAN DARE 2	45891,0 – жизнь
SCEPTRE OF BAGDAD	58116,0 – жизнь
REBEL STAR 2	28599,N - кол-во ходов
PHANTOM CLUB	56486,0 - жизнь
TRUENO 1	25100,N - энергия
TRUENO 2	24785,0 - жизнь
	24984,N - энергия

Серию подсказок, достаточную для того, чтобы пройти всю игру SCEPTRE OF BAGDAD прислал наш читатель из Каракашев А.Г. из г. Грозный. Он очень интересуется адвентюрными играми, но, как и многие, не имеет возможности пополнять коллекцию в связи с тем, что они пока мало распространены.

1. Имея каску, можно пройти в комнату, где должна быть женщина, закрывающая ход в спальне. Теперь каску можно выбросить, больше никогда никто здесь не появится.

2. Флейта нужна, чтобы моток веревки при Вашем появлении в комнате поднимался вверх - тогда по ней можно подняться.

3. Имея крылья, идите к фонтану, ангел улетит на небеса, откроется подземный ход.

4. Огненного круг откроет люк (в фонтане).

5. Со связкой ключей идите в комнаты, расположенные за проходом, который раньше закрывала малоприятная личность, боящаяся каски. Дойдя до конца, прыгните в шкаф, ключи откроют потайной ход.

6. В фонтане трезубец можно обменять на жемчужину.

7. Имея сачок, идите в комнату с пчелой (пчелу предварительно надо выпустить из гнезда). Поймайте пчелу и отнести ее в комнату с пауком. Паук исчезнет.

8. Шпагой можно отрубить веревку, на которой висел паук.

---

[1] Скорее всего 561 (Прим. NUK)

9. Жемчужиной, взятой в фонтане, зарядите рогатку.

10. Заряженной рогаткой сбейте кокосовый орех с пальмы, где сидит обезьяна.

11. Имея кокос можно идти в жаркую пустыню.

12. С книгой Али-Бабы можно открыть потайной код в подвале.

13. После этого можно в подвале наполнять кошелек золотом.

14. Имея кошелек, можно зайти в магазин и купить топор и башмаки.

15. Хлыстом усмирите быка и обменяйте его на ось.

16. Ось замените в подвале на сломанную.

17. Топором заточите кусок дерева.

18. С помощью зеркала убейте медузу.

19. Заточенным деревом убейте людоеда.

20. Веником усмирите волшебника и возьмите лампу, с которой можно смело идти за скипетром.

21. Одев башмаки, можно пройти через раскаленные угли.

22. Имея рыбу-шлем можно дышать под водой.

23. Веревкой натягивается лук.

24. Заряжается он стрелами.

25. С заряженный луком можно идти в комнату, расположенную за людоедом.

Далее уже совсем легко. Взяв жезл, возвращаетесь на балкон справа от исходной комнаты. Теперь мы видим счастливое лицо нашего героя, сжимавшего в руках заветный скипетр.

Четыре предмета: песочные часы, тряпку, доспехи и перо применить нигде не удалось, но видимо они и не нужны, т.к. было набрано 100% очков.

Некоторые полезные советы:

1. Последовательность выполнения действия любая, удобная для игрока, правда, пройдя игру несколько раз, можно попробовать искать все более короткий путь. Конечно, при первом проходе трудно избежать бестолкового шатания по всему лабиринту туда-обратно.

2. Старайтесь никогда не делать себе безвыходных ситуаций, например, можно, забравшись по веревке вверх, умудриться забыть там флейту и спрыгнуть вниз без нее, или, например оставить в магазине мешок с золотом и выйти. Это может свести на нет все Ваши усилия.

3. Найдя бутылку, дающую OLD GAME не спешите тут же хватать ее, если у вас еще достаточно жизней, а старайтесь сделать как можно больше действий с предметами, и взять бутылку, только когда жизней остается опасно мало.

Внимание!

Дорогие друзья, сегодня кы ножен помочь Вам стать непосредственными участниками новой, гото-вяяейся в настояиее время телеви-зионной игры, мы уполномочены на то, чтобы раскрыть перед Вани некоторые детали этого мероприятия и пригласить Вас к участив в небольшом конкурсе.

Ее устроитель. фирма •фонд", поставила перед собой интереснейшую задачу объяснить в Форме популярной телевизионной игры что такое акции, что такое Фондовая биржа, что такое дивиденд и как образуется курсовая разница. Зрители смогут наглядно увидеть и почувствовать, что такое фондовый рыбок, узнают многое из того, что поможет им в будушем из бежать возможных неприятностей, связанных с вложением денег в ценные бумаги.

"инфорком" участвует в подготовке этой телепрограммы на некоммерческой основе, обеспечивает ее компьютерную поддержку и, до некоторой степени, участвует в выработке ее концепции.

Начало регулярных трансляций телеигры в эфире запланировано на август-сентябрь несян сего года. Названия телеигры мы Вам пока сообщить не можем, оно проходит регистрацию.

Теперь о сути.

организация игры достаточно традиционяа: пять игроков на спе-не и несколько рядов зрителей в студии. игроки получают начальный капитал (5000 рублей) и могут им распоряжаться, вкладывая деньги в акции тех или иных компаний. Акции можно не только покупать, но и продавать, если Вы чувствуете, что над компанией нависла угроза.

Всего в игре могут быть задействованы акции следующих двадцати компаний:

судостроительная;

- лесоперерабзтываюяая;
- Торговый Дом;
- биржа;
- пиаевая.
- авиационная;
- Фармацевтическая;
- трансаортная;
- топливно-энергетическая;
- газовая;
- пивоваренная;
- оборонная;
- банк;
- станкостроительная;
- золотодобывающая;
- прохладительных напитков;
- строительная;
- полиграфическая;
- автомобильная;
- жяншческая.

Финансовое положение этих компаний нестабильно и стоимость их акпий непрерывно изменяется, предугадать, как поведут себя акции той иди иной компании можно только если очень внимательно читать газеты и делать при этом правильные выводы из прочитанных с со биений.

Перед каждым кодом игрокам дается какое-то газетное сообаенне, которое может быть и совершенно шуточным и абсолютно серьезным, а уж они сами должны сообразить. как оно повлияет на стоимость акций тех или иные компаний.

Давайте рассмотрим, например вот такое сообщение:

•Как сообщают из высокопоставленный кругов, близких к правлению корпорации "Детский нир", в течении двух-трех ближайших недель готовится десант Санта-Кяа-усов из

штата Техас, в качестве ближней тактической задачи перед десантной группой ставится поз дравление воспитанников московс ких детских домов и школ интернатов. Десант возглавляет президент компании "Шеврон", "

такое сообщение может иметь. например следующие воздействия на рынок ценных бумаг:

торговый дон '«ю/

Биржа »гох

Фармацевтическая MS/ Топливно энергетическая *?.2/

Газовая мех

Пивоваренная ,0/.

Банк »15 у.

Прохлад, напитков 25/

Автомобильная - п х

Теперь рассмотрим, в чем здесь суть. в этом сообиении две ценных информации.

Во-первых, это привязка по дате. Очевидно, что речь идет о на -чале декабря месяца и приближается Рождество со всени вытекающими отсюда последствиями.

Во-вторых Си это более сложно) те, кто внимательно глелит за экономическими новостями, знают. что компания "Веврон" подписала соглашение с н. Назарбаевым о совместной эксплуатации нефтяных по лей в Прикассин. эти нефтяные поля являются совместными для Ка эахстана и россии и можно вредно дожить, что гуманитарный визит президента этой компании в москву ножет в будущем привести и к заключению аналогичных соглашений и с Россией.

Итак, акции торгового дона имеют всплеск, потону что в последний несяп перед Новым Годом совершается покупок больше, чем в последующие три месяца. Все хотят поздравить родных и близких, кроме того, население, приученное к тону, что в новом году обязательно будут новые пены, спешит ку-

пить все, что запланировано, в последние недели.

возрастание общего объема продаж влечет и повышение деловой активности на бирже и рост доходов биржи.

Временный подъем ножет иметь и Фармацевтика, ведь в этот осенне-эимний период резко увеличивается количество вирусных заболеваний. а что делается в дни школьных каникул после многочисленных "Ново годних Елок" не надо объяснять.

Газовая промышленность имеет рост. поскольку приближаются зимние холода и требуется повышенный расход газа. То же относится и к топливно-энергетическому комплексу, но Добавим erne ранее ткка-занкые предположения о цеди приезда пре зидента компании "мев

рон".

Обшии рост деловой активности несколько поднимает и акции бан ков.

Спад имеют автомобильная промышленность (спрос на автомобили в начале зимы традиционно меньше, чей в начале лета) и. конечно, сокращается потребление пива и прохладительных напитков. I!

Ксли хотите, можете попробо вать свои силы в ш>илумывании по добных сообщений. Присылайте то. что Вам удастся изобрести вмгтт<> го своими предположениями о j>i\iic аим рынка на это сообщение. н

фирма -фонд" будгт ржемггичноЦ отбирать 7 8 лучших гоо(>воний длив включения их п последующие игры,и а авторы этих сообщений получат! приглашения в студию и качествов зрителей. иногородним ФирмЛ "фонд" обеспечит проживание л компенсацию затрат ил проегт. п

Кроме этого, срели приглашен S ных будет организован блиц тур I нир. Кго победитель получит право! занять пятое игровое место mjj сцене и безвозмездно получит ис I

ходкую сумму в 5000 рублей, н"0<. 1

ходиную для участия л игре в ко ле игры у него будет возможность ее увеличить или уменьшить

После выхода в теле:)Фир ирг зентапионной передачи 1яргугт сентябрь) к участию в этой кон курсе подключатся миллионы теле зрителей, но пока вы. напи ува жаеные читатели, имеете неоспори мое преимущество.

Хелаем Вак успеха!

Свои варианты присылайте пока на наш адрес, мы передадим их устроителям телепередачи • Фирме "фонд". На конвертах делайте пометку "TV".

Уточняем наш адрес:

121о19, Носква, г-19. а/я 16.

Вниманию наших коммерческих представителей на местах, а также всем желающим подключиться к Распространению программных продуктов "ИНФОРКОМа" для IBM-СОВ- і иестимых компьютеров мы представляем обучающий программный КОМПЛеКС "барьер".

J. назначение конплекса

Комплекс предназначен для развития навыков скоро- : чтения и, тем самым обеспечивает повышение интенсив- і нести усвоения любых текстовых учебных материалов. <

комплекс состоит из трех программ и имеет широкий і спектр модификаций. Разработанный в период октябрь 199о - мзрт 1991г. . он реализуется нами на коммерческой основе уже более года. Количество продаж на сегодняшний день приближается к одной тысяче. в порядке і подведения первых итогов скажем, что комплекс собирает самые благоприятные отзывы, наиболее мошшй эффект ; он имеет для детей любого возраста (умевших читать).

Программы сами настраиваются на уровень подготовки пользователя, который с ними работает, не требуют ни- і каких навыков по работе с эвн. имеют ярко выраженный і соревновательный характер и могут рассматриваться не і только как тренирующие, но и как тестируйте и как : игровые.

барьер-1. Программа предназначена для развития г навыков "Фотографического" восприятия текстов и интенсивно тренирует Фотографическую память.

карьер-2. Программа предназначена для развития пе- : риферического зрения. с помощью тестовых таблиц про- ' изводится диагностирование распределение концентрации внимания по зрительному полю. I

барьер-3, Тренирует блочное восприятие текстов при) неполной информации (когда левая и правая часть поля Нстранипы не охватываются периферическим зрением).

в II. комплект поставки

комплекс поставляется на трех дискетах 5.85" (MS DOS. 360 к). Каждая программа на отдельной дискете.

III. модификации комплекса Программы комплекса "барьер" дают проверенный зФ-Фект независимо от того, кто с ними работает, но тем не менее, мы подготовили ряд параллельных модификаций, рассчитанных на то, чтобы учебно-тренировочные текс.-|-ы, используемые в программах, были по-возможнос-ти максимально приближены к конкретной сфере деятельности нашего заказчика.

это следующие модификации: барьер-н - менеджмент и маркетинг II барьер-в - вычислительная техника и программирова-I! ние.

и барьер-д - для детей (младший школьный возраст). II Эта версия адаптирована для детей в ![сиысле особого подхода к подбору учеб-II них текстов. Как показывает опыт эксп-II луатации, одновременно с значительным (в г-3 раза за один несяп) возрастанием скорости чтения у детей развивается нанять и правописание. Эта версия пользуется наиболее широкой популярностью. барьер-п - автоматизация производства и проектирования в машиностроении.

барьер-т - металлообработка (технология машиностроения)

барьер-и - металлообработка (станки и инструмент) барьер-р - радиотехника и электроника барьер-С - сельское хозяйство (растениеводство) барьер-о - обшекультурное содержание, эту версию заказывают при неопределенной профессиональной ориентации, например для школьников старших классов или для тех, кто связан с гуманитарными областями деятельности, может быть рекомендована клубам, кружкам и т. п.

примечание: Если в заказе не указано конкретно, какая версия необходима заказчику, мы поставляем версию барьер-в (вычислительная техни-| ка и программирование)._____

IV. срок исполнения заказа.

срок исполнения - е - 3 недели после поступления средств на наш р/с.

V. технические требования к аппаратно-программнону

окружению

1. Полная аппаратно-проГРамнвая совместимость с IBM PC XT/AT, надежность функционирования на отечественных модификациях не гарантируется и не обсуждается.

а. Наличие "жесткого" диска ("Винчестера") стандартного объема.

3. Дисковод гибких дисков 5,35".

4. Операционная система - HS DOS не ниже э. 20.

5. Русификация компьютера в стандарте гост (кодировка альтернативная).

6. Требования к монитору - не специфицируются. Желательно - EGA.

VI. гарантийные обязательства

гарантийный свидетельством при поставке программного продукта является картонный альбом, в который вложены дискеты с указанной на иен датой продажи. при его отсутствии поставка выполняется с заверенный гарантийный талоном.

гарантиями обеспечивается:

- бесплатная замена поставочных дисков, неработоспособных в состоянии поставки (в течение месяца после поставки);

- замена с минимальной оплатой ори выходе программ из строя по вине пользователя (механическое или злек троиагнитное повреждение, поражение вирусом на машине пользователя и т. п.) или по истечении месяца после поставки. Минимальная оплата не превышает стоимость дисков * 5х текущей стоимости прогрэнного обеспечения + стоимость почтово-транспортных расходов и согласовывается с потребителей.

VIII. порядок оформления заказа.

а) направить в наш адрес письмо-заказ с указанием необходимого программного продукта и количества копий. приложить копию платежного поручения.

Крайне желательно сообщать также номер банковского авизо. Это позволяет проводить розыск перечисленных средств в случае их длительной задержки в структурных подразделениях центрального Банка. Наш адрес: 121о19, Москва, г-19, а/я 16, "инфорком-.

б) произвести предварительную оплату платежным поручением на наш р/с: и 5о04617т8 во фрунзенском коммерческой банке г. Москвы. мфо гО1412.

Стоимость комплекса на период май - август 1992г.

барьер-н юо руб. + гвх

барьер-в - 100 РУб. * 28/

барьер-д - 100 РУб. + 28х

барьер-п - 100 РУб. * 28/

барьер-0 - 100 РУб. * Jвх

барьер-т - 300 РУб. * 28х

барьер-и - 300 РУб. * 28'/

барьер-р - 300 РУб. * 28Z

барьер-с - зоо РУб. * 28х

"зеленый пакет" дистрибутора

о том, что такое "зеленый пакет" Вы можете подробно прочесть в ни-12 "ZX-ревю" за 1991г.

стоимость "зеленого пакета" по комплексу "барьер" составляет для частных яиц 430 рублей, высылается наложенным платежом. Никакой предоплаты делать не надо, достато^шо прислать заявку.

напоминаем, что затраты дистрибутора на "зеленый пакет" являются только залогом и возвращаются ео требованию. Активно работаю»» дистрибуторам затраты возвращаются при выплате комиссионных и далее такие пакеты предоставляются бесплатно.

мкп "инфорком" 121019, Москва, г-19, а/я 16

Уважаемые читатели:

Ны хотели бы обратиться к вам со следующей информацией. Просим ею руководствоваться в дальнейшей деятельности.

L. Иы определились в наших планах на 1993 год и продолжим выпуск "ZX-РЕВО" в тон хе виде, т. е. это останется все такое же внутреннее "Фирменное• издание для своих клиентов. Правда, по понятным экономический причинам тираж его будет еще более сокращен.

г. Ны не снохен принимать от вас подписку по почте так. как это делалось в прошлые годы и организуем в Москве корреспондентский пункт. Подписаться смогут только те. кто лично придут по известному им адресу или пришлют своих друзей, находящихся в

Москве проездом, подробности о тон. как связаться с корпунктом -в прилагаемом к данной? выпуску информационной листке.

3. в порядке исключения для жителей особо удаленных районов, не имевшим никакой воэнохности прибыть на корпункт мы рассмотрим проведение подписки по почте, хотя гарантируем это не всем. Будет отдано безусловное предпочтение нашим постоянным клиентам и авторам.

4. на корпункте кроме подписки на 1993 год мохно будет приобрести выпуски 1992 года, полный комплект 1991 года, выполненный в виде отдельной книги улучшенного качества и прочие сопутствующие материалы.

5. стоимость подписки и прочих материалов, при приобретении их через корпункт будет низе, чем при аналогичном обслуживании до

почте.

6. Мелкооптовые ооераюш будут обеспечены дополнительными льготами.

7. Ны заключили соглашение о сотрудничестве с редакцией журнала •монитор", согласно которому часть материалов этого многотн-рахного журнала, посвященная особо слохным и интересным игровым программам для IBM-совместимых машин будет готовиться нами, начиная с сентября месяца этого года IN5) и далее ежемесячно наши материалы будут печататься в этом журнале. Рекомендуем это издание Вашему вниманию как для подписки, так и для размещения рекламы.

журнал "нонитоР"; I Тираж зо 000 экз. I Периодичность в 1992 году - 9

номеров в год.

Издатель: нто "СоФт-носква". Основная Форма распространения! розничная продажа, но подписка по почте вознохна. I

Адрес для писем: 117419. носква. а/я 765. Тел. 247-36-35; 137-21-36.

Сообщаем вам график выхода очередных выпусков zx-ревю этого года (по началу рассылки): н7-в - 20 сентября N9-10 - 30 октября н11-)2 - 15 декабря До свидания, до встречи в следуюшен выпуске.

ваш инфороком.

СПЕКТРУМ В ШКОЛЕ

В прошлом выпуске "ZX-РЕВЮ" мы напечатали тестирующую программу, которая может быть полезной на уроке истории. Сегодня мы приводим рекомендации по тому, как ножно использовать компьютер на уроке географии.

Перед учащимся появляется на несколько секунд карта какой-либо страны (области, края) с указанный расположением городов. Затем города исчезают и остается только контур страны. Учащийся должен курсором указать где находится тот или иной город.

вы можете сами задать ту иди иную карту (в ваяем примере рассмотрена Австралия). Это может быть, например Красноярский кра$.

Не надо думать. что эта программа применима только для определения несторасположения

городов. С тем же самым успехом можно проверять знание учащимся основных районов добычи тех или ияых полезных ископаемых, знание расположения ГОРНЫХ хребтов, названий рек. расположения гидро-комплексов и т. п. Вы сами легко разберетесь, как Вам развить и применить эту программу.

Преподаватели биологии сяргут применить аналогичный подход для проверки правильности знания учащимся названий различны» частей растений или скелета животного, возможностей иного. Ны надеемся, что Вы сумеете ими воспользоваться.

```
10 REM Урок географии
20 LET another ~- 150:
LET nextclty = 300:
LET tryaeain = 380:
LET move = 400:
LET wait -- 420:
LET finish = 580:
LET outline - 1000:
LET PrintCltT - 2500:
LET test - 27QO:
LET data = з000
30 RBi ИНСТРУКЦИИ обучаемому
4О ВЖ1Ж l: PAPER 7: IEE 9:
BRIGHT l: CLS 5О PRINT PAPER l; FLASH 1; AT
9,9;'УРОК ГЕОГРАФИИ-
60 PAUSE 150
70 CLS 80 PRINT AT 2.3;
"Эта программа проверит Ваши знания по географии Австралии, на десять секунд Вам будет
      показано расположение австралийских городов. Потом они исчезнут и останетсяй
      только КОНТУР. Ваша задача | установить указатель в том п м«сте. где должен
      находиться I заданный Вам ГОРОД. I Вы имеете по три попытки на I отыскание
      каждого города. " I 90 PKINT AT 19,5; FLASH 1; чЬжните любую клавишу
100 ftAUSE О
110 Ctfi 11Ф «INT AT 2. 8;
•Клавиша в - указатель вправо клавиша 5 - указатель влево Клавиша 7 - указатель вверх
      Клавиша 6 - указатель вниз Нажмите О, когда указатель займет правильное положение
      после каждой попытки курсор будет возвращен в левый
шцшмй угол. •
150 PRIBT FLASH 1:ЛТ 19,5;
"Нажните лвбтв клававу"
НО PAUSE О
150 KEM another 160 DIB r(4> 170 LET error = o 180 GO SUB outline 190 RESTORE data 200
      FOB n=0 TO 6
aio READ a. b. Ґ. X. a$ гго GO SUB printcity r 30 NEXT n
840 RESTORE data гво PAUSE 500
J90 GO SUB outline
300 REM nestclty
310 LET SClty = 0: LET yclty = 0:
LET tries : l 320 LET s* = " ": REM десять пробелов
330 PR1HT AT 1.1:9*
340 PRINT AT 16. 1: "Найдем-";
AT 19. 1:9» 350 READ a, b. Ґ. X, a$ 360 IF a$ = "eof- THE» GO TO
finish
3TO PRINT AT 19. 1: at 380 REM tryaeain 390 PRINT AT l. i: -attempt •;
tries
400 REM move
410 PLOT OVER 1: xcity. yclty 420 REM wait
430 LET fall = o: LET result = 0
```

```
440 IF 1HKEУ4 - '0" THEM GO SUB
test 450 IF result - г THEN
LET runes) =r(tnes) »1:
PRINT FLASH 1;AT 20, I;
•CORRECT": PAUSE 25:
PAUSE 150: PRINT AT 20. 1: s»:
GO SUB prlntcity 460 IF tries - 4 THEN
LET П4) =r[4) + l:
LET error - l:
GO SUB prlntcity 470 IF result - 1 OR tries = 4
THEN 80 to nextClty 480 IF fail = 1 OR tries = 4 490 LET dg=UKKEY» = "8")
-<INKEY»="5"> 500 IF BCity*dK=256
OR scity»dx=-l
THEN LET dX'-O 510 LET dy; <IKKEY»="7">
-{[HIJY»="6"> 520 IF ycity«dy=176
OR yclty*dy=-l
THEN LET dy-O 530 IF dx=0 AH» dy--0
THEN GO TO wait 540 PLOT OVER 1; XClty. yclty 550 LET XClty=XClty*dx 560 LET
      yclty=ycity«dy
570 GO TO move
580 REM finish
590 CLS
600 PRINT AT 4, 2;
"Верно с первой попытки: ";r(li 610 PRINT AT T, 2; "Верно со второй попытки: ";г(2)
620 PRINT AT 10.2;
"Верно с третьей попытки: -;г(3)
630 PRINT AT 13.2;
"Неверно: ":r(4J
640 INPUT -Попробуем еще раз?". У*
650 IF CODE у»=89 OR CODE У* -121
THEN GO то another
660 STOP
950 КЕН'""'»*""""»
• Подпрограюш •
к •
1000 КЕН КОНТУРЫ
1010 CLS ЮгО PLOT 51.58
1030 DRAW 0.4
1040 DRAW 2.0 1000 DRAW 0.9
1060 DRAW -15,31
1070 DRAW 4. -3
1080 DRAW 1, 2
1090 PLOT 51. 128
1100 DRAW -8, -8T.1. 5
1110 PLOT 51. 126
1120 DRAW 3, -1
1130 DRAW 6,3
1140 DRAW 12.9, 1. 5
1150 DRAW 3. 1
1160 DRAW -1.4
1170 DRAW 3.5
1180 DRAW 3.-6
1190 DRAW 2.3
1200 DRAW -2.2 1210 DRAW 1.2 1220 DRAW 4, -J 1230 DRAW 0, 7 1240 DRAW 10,7 1250 DRAW 3.-
      1 1260 DRAW 3. -5 1270 DRAW 4, -2 1280 DRAW -2.4 1290 DRAW 7,9 1300 DRAW 8,1.1
      1310 DRAW 0.J 1320 DRAW 2,1 1340 DRAW 1.-2 1350 DRAW 19.0. 1 1360 DRAW 2. -2 1370
      DRAW -7, -13 1380 DRAW 21, -14 1390 DRAW 4. 1 1400 DRAW 5. 11 1410 DRAW 2.20 1420
      DRAW 3. 1 1430 DRAW 1.-6 1440 DRAW 2.-4 1450 DRAW 1, -9 1460 DRAW 4. 1 1470 DRAW
      0, -E 1460 DRAW 3, -3 1490 DRAW 0. -7 1500 DRAW 2. -2 1510 DRAW 2,-10 1520 DRAW
      2. -1 1530 DRAW 11,-13 1540 DRAW 0, -5 1550 DRAW 3,0 1560 DRAW 0,2 1570 DRAW 3. -
      1 1590 DRAW 0, -4 1590 PLOT 210.71 1600 DRAW -6. 37. 1. 5 1610 PLOT 210.71 1620
      DRAW -15. -26 1630 DRAW -3. -6 1640 DRAW -2.-1 1650 DRAW -8. -3, 1. 2 1660 DRAW -
      2.-1 1670 DRAW -I.-3 1680 DRAW -2,3 1690 DRAW -7.4 1700 DRAW -5, -3 1710 DRAW -
      11,4 17ЙО DRAW -2,5 1730 DRAW 1. 1 1740 DRAW -5.7
```

```
1750 DRAW -2.-1 1T60 DRAW 0. 8. 1 1TTO DRAW -3. -5 1700 DSAW -1.5 1790 DRAW 3.3 1B00
      DRAW -2.3 I810 DRAW -9. -9 1820 DRAW -JO, 5, 2. 4 1830 DRAW -13,-2 1840 DRAW -6,
      -3 1850 DRAW -7, -1 I860 DRAW -6.4 1670 DRAW -l.-l 1««0 PLOT 213.96 1690 DRAW
      2,4,. 5 1900 PLOT 142.55 1910 DRAW -2,0 1920 DRAW -1.-2 1930 DRAW ?.,0 1940 DRAW
      1,2
1950 PLOT 170.25 1960 DRAW 15.-1.. 7
1970 DRAW 1,2
1960 DRAW -1.3
1990 DRAW 3, -2
2000 PLOY 163.9
2010 DRAW 5.16.. 5
2020 PLOT 183.9
2030 DRAW -3.0
2040 DRAW 0. -2
2050 DRAW -4.0
2060 DRAW 0. 1
2070 DRAW -2, 0
2080 DRAW -1. 10
2090 DRAW -3.4
2100 DRAW 0.4
2110        RETURN
2500 REM ГОРОДа
2510 PLOT FLASH error;a.b
2520 PLOT FLASH error: a. b+1
2530 PLOT FLASH error;ач.ь 2540 PLOT FLASH error; a$l. D»l 2550 CIRCLE FLASH error; a,
      ь*1. з
2560 PRINT FLASH error: AT V. X: a$ 2570 IF erroГ:0 THEN            RETUR8 2580 PAUSE
      50: PAUSE 200
2590 LET error = 0
2600 GO TO prlntcity
2700 REM test
2710 PLOT OVER 1: «city, yclty
2720 IF ABS (XClty-a) < 4
            AND ABS СУС11У-Ы < 4
THEN LET result ; 1 2730 IF result = 0
THEg LET trles=trles*l: LET fall=l
2740 LET xclty = 0: LET yclty = 0
2750 PAUSE 25
2760        RETURN
2900 REM ••«•••»•«••>
• Данные • з000 REM Damoie по городам
3010 DATA 172.39.17,22.
"Мельбурн-
30EO DATA 199.56.15.25.
•сиднев-зозо DATA 53, 71, 13,7.
"Перт-3040 DATA 147,55.13.14.
-Аделаида"
3050 DATA 110.165,1.14.
-Дарвин-
3060 ПАТА 179.10.20.23.
"Хобарт"
3070 DATA 212. «1. 10. 23.
"Брясбеяя" 3080 DATA 0. 0. 0. 0. 'eof"
```

BETA BASIC

Продолжение, начало см. стр. 3. 4Т.

16. DO

или во WHILE <условие>

или DO UHTIL <УСвовне>

клавиша: D (Ключевое слово WILE находятся на клавише J, а ключевое слово USTIL - на клавише E1-

Сн. также LOOP. EXIT IF.

конструкция DO - LOOP инеет ряд преимуществ по сравнению с обычным способен организации циклов FOR - NEXT стандартного бен-C8Ka. эти преимущества становятся еве более ояутинынн при непользе-вавхн квалиФккаторов VHILE и

UHTIL.

Без ниж DO и LOOP являются просто маркерами, отмечавшим начало и конед пекла. После того. как программа встретит оператор LOOP, травление передается на оператор во и т. д.

Пример:

```
10 DO
20 PRINT 'HELLO "
30 LOOP
```

Эта программа будет печатать бесконечное число раз слово HELLO. Остановить ее можно будет только нажав BREAK.

Действие DO можно изменить с помощью квалифнкатора WHILE <ус-ловие>. Его действие таково:

Если условие стояке после WHILE справедливо (имеет значение "истина"), то выполняются операторы, стоящие после DO до тех пор, пока не встретится оператор LOOP, после чего управление вновь передается на DO я вновь проверяется справеливость условия и т. д. Если же условие "ложно", то вся часть программы, стоящая между DO и LOOP игнорируется и управление передается к оператору, стоящему за LOOP.

Таким образом, та часть программы, которая с тоит между DO WHILE <УСЛОВИе> и LOOP выполняется раз за разом, пока <условие> справедливо.

DO UHTIL <условне> имеет прямо противоположное значение, часть программы, заключенная между DO и LOOP выполняется, пока условие "ложно". (Иными словами: до тех вор. пока условие не станет справедливым).

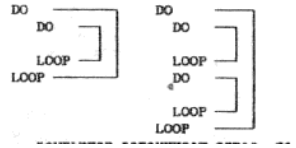
Пример:

```
10 LET total - o
80 DO U1TIL total > 10O
30 IKFVT "Введите число "; x
40 LBT total = total + x
So PKIVT total
GO LOOP
TO PKIKT -получили число болым ста"
```

в этом примере строку 𝔍0 можяо было бы заменить таков:

```
20 DO VHILE total <- 100.
```

Пары DO-LOOP могут вкладываться точно так же. как FOB - NEXT. Например:



Компьютер запоминает адрес, по которому в программе хранитея оператор DO. для запоминания служит стек. поэтому нельзя выходить из цикла (перепрыгивать через LOOP), иначе как с помощью EXIT IF или POP. Иначе может произойти "закупоривание" стека и с бой работы в программе.

Если в программе для оператора DO не найден соответствующий LOOP. выдается сообщение об ощибке: S. 'Hissing LOOP-.

Аналогичные конструкции существуют и во многих других языках программирования и могут иметь другое написание, вот примеры аналогичных конструкции:

REPEAT
операторы URTIL <усдовже> аналогично: DO
операторы LOOP UHTIL <условие>
REPEAT
операторы UHTIL FALSE
аналогично: DO

оператор» LOOP
WHILE <условие>
операторы
ENDWHILE (или WEED) аналогично:
DO WHILE < условно
операторы
LOOP

17. DPOKE адрес, число

Клавиша: р

См. также DPEEC (адрес),число

DPOEE означает "двойной" POKE.

Этот оператор засылает двухбайтное число в две адресных ячейки. Аналогичная конструкция а стадартом БЕВСОе имеет следующий вид:

росе а. п - 11т(й/25бН25б

роке а+1.11т<п/25б) здесь а - адрес

п - число со. ..65535)

другими словами, нланшни байт засылается по указанному адресу, а старший байт - в следующий за ним адрес.

Поскольку многие системные переменные "Спектрума" имеют именно такой двухбайтный формат, то их очень удобно изменять с помошью DPOKE. Аналогично функция DPEEK представляет "двойной" PEEK.

18. DKAW то X. г <.угол>

Здесь используются два ключевых слова стандартного БЕИСНКа -

DRAW и то.

Этот оператор вычерчивает линию от текущей графической позиции до точки, заданной координатами к, у. Часто это более удобно, чем использовать стандартный DRAW, при котором задаются не абсолютные координаты, а величина относительного "сневенмн" по горизонтали и вертикали.

Пример:

```
10 FOR n=1 то 1оо
20 DRAW TO RKD"255,RND'17b
30 NEXT n
```

Если Вы зададите и третий параметр (угол), то сможете изображать кривые.

После то вн можете поставить оператор цвета PAPER. евс или атрибут, например OVER и т. п.

DRAW TO 10. 10

DRAW TO IKX 2; 20. 30

DRAW то 100,90, I

18. EDIT <номер строки>

Клавиша: о (в обычном, ке графическом режиме).

Это не то же самое, что SHIFT * "1", EDIT - это ключевое слово. оно предназначено для того, чтобы избежать утомительной последовательности: ызт - выбор строки -BREAK - SHIFT + "г.

Для того, чтобы быть по-настоящему полезным. EDIT должен вызываться без нажатия SHIFT клавиши. Но все буквенные клавиш ухе заняты, а цифровые клавиши нужны для ввода номеров строк. в то же время, номера строк не начинаются с нудя. поэтому эту клавишу можно использовать.

Вы закончили набор какой-либо строки, нажали ентек. программа ждет ввода номера очередной строки. Вы нажинает о и получаете ключевое слово EDIT (если даже оно и не появится, все равно нудь в начале строки означает EDIT). Если после EDIT (или нуля) набрать номер нужной Вам строки и нажать кхткн. эта строка мгновенно будет помещена в нижнюю часть экрана для последующего редакти-

рования.

Если номер строки не указан, то на редактирование поступит текущая строка, дополнительно для упрощения редактирования введена возможность перемешать курсор по экрану "вверх"/"вниз" в рамках программной строки. курсор не нохет вставать внутри ключевых слов, а занимает блихайший пробел. Если Вн попытаетесь поднять его вине, чем вершина строки или опустить ниже нижней строки экрана, он автоматически "прыгнет" в конец программной строки, самый простой способ добавить что-то к концу строки - вызвать EDIT и затем нажать "курсор вверх".

19. EDIT строковая переменная

или EDIT ; числовая переменная

Клавиша: SHIFT *- "5"

EDIT может применяться не только для удобного редактирования строк, но и для простого изменения программных переменных. Для этих целей нельзя использовать нудь в начале строки и введена возможность вызова EDIT нажатием SHIFT + "5" в графическом режиме, Можно также набрать слово EDIT по буквам в режимах KEYWORDS з или 4. Типичное применение -внесение изменений в строковые переменные, например для изменения фамили. имени отчества, адреса в массиве данных', представляющем из себя базу данных.

Такой массив мог быть заполнен, например с помощью INPUT. Если теперь надо в нем что-то изменить, то это трудоемкая задача для обычного БЕЙСИКа. а здесь с помощью EDIT a$ <п) Бы получаете содержимое строки 'п' в области редактирования, Рассмотрим пример:

```
10 LET a$ = "John Brown"
20 EDIT a$
30 PRINT a$
40 LET nUBD=365. 253
50 EDIT ;лш)
```

60 PRINT num

в строке 50 применен знак ";" для того, чтобы отличить режим EDIT (числовая переменная> от режима EDIT «номер строки>. Нож-но было бы. однако, вместо точки с запятой использовать и запятую, но в этом случае редактируемая переменная была бы изображена, начиная с 16-ой позиции экрана.

EDIT имеет синтаксис, очень ипуишй на iflpur. Бы можете использовать точку с занятой, запятую. AT. TAB, LIHE и строкОВУЮ

подсказку точно так же, как обычно это делают с оператором INPUT, Отличие в том. что редактировать можно только одну переменную. Если Вы попробуете редактировать несколько, то все прочие, кроне

первой, будут восприняты. как операторы INPUT. Нижеследующий пример показывает, как можно создавать массивы и редактировать их:

10 DIM a$(10, 15) 20 FOR п=1 то 10

зо INPUT a$cn)

ад NEXT п

50 PRINT "Редактирование*

60 FOB n=l TO 10

то EDIT ("завись ";п:" ">;а»(п) 60 NEXT п

Если переменная. подлежащая редактирование в EDIT не существует, то команда полностью эквивалентна INPUT.

20. ELSE <оператор>

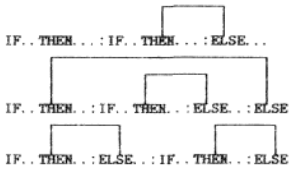
Клавиша: е

ELSE - один из элементов конструкции IF - THER. Обычно, если условие, стоящее после IF несправедливо (ложно), то следующей выполняется строка, стоящая за строкОЙ, содержащей этот IF. этот порядок можно изменить. Если IF и THEM содержат еще и ELSE, то в случае, когда условие не выполняется, управление передается оператору, стояшену после ELSE. с другой стороны, если условие справедливо, то строка IF исполняется только до ELSE и затем управление передаете я следующей строке (если тнен не передал его в иное место). Например: 10 INPUT "Задйте число ";x 20 PRINT "Это число равно 1?" 30 PAUSE 50 40 IF x-1

тнен PRINT "да": ELSE

PRINT "нет" 50 GO TO 10

однако. ситуация может быть осложнена, если на одной строке у вас есть несколько IF тнен ELSE. Тогда может быть трудным определить, что же к чему относится. Приведенные ниже примеры помогут Вам разобраться.



Если ELSE используется без IF, например, когда ELSE стоит первый оператором строки, то и ELSE и остальная часть строки игнорируются.

21. END PROC

Клавган: 3

См. также раздел, несведённый процедурам. PROCEDURES; DEF PROC.

Этот оператор отмечает конец процедуры. Благодаря этому ком-

пьютер не будет выполнять ничего, стоящего между DEF рвос и END ряос. если процедура не была вызвана, он просто перепрыгнет через нее за ею) рвос. Во время же исполнения процедуры. EHJD ркос служит концом ее «сволнения. Удаляются локальные переменные (LOCAL) восстанадиваются исходные значения глобальных (GLOBAL) переменных, если они есть. Если в процедуру какие-то параметры передавались по ссылке (перед Формальным параметром стояло RKF). то текущее значение формального параметра присваивается Фактическому параметру, задействованному при вызове.

Если Формальный параметр не найден. Вы получите сообиение "Variable not found*.

после исполнения процедуры управление передается оператору, стоящему за вызовом процедуры.

применение END F80C без соответствующего DEF PROC дает сооб-шение об ошибке:

W, "Missing DEF PROC".

22. EXIT IF <условие>

клавиша: I

См. также DO. LOOP

этот оператор является элементом конструкции никла DO-LOOP <см. соответствующий раздел). EXIT IF используется для того, чтобы выйти по своему желанию из середины цикла DO-LOOP, если условие выполняется. После выхода из цикла управление передается оператору, следующему за LOOP.

Если условие не выполняется, ничего не происходит.

Пример:

```
100 DO
по P8INT "строка но-
120 PAUSE 20
130 EXIT IF IRXE7* = " STOP "
1вд PRINT "строка 140"
150 PAUSE 20
160 LOOP
170 PRINT "Выход из цикла"
```

Программа будет работать до тех пор, пока не будет нажата клавиша "STOP" (sm SHIFT * a).

Если в программе опушен оператор LOOP-, выдается сообщение об ошибке: S, "Hissing LOOP".

23. FILL x,y

иди FILL <INK число!> X,г

или FILL <paре8 число;> x, y

Клавиша: F

Команда FILL или FILL IHZ закрашивает область экрана, окрашенную цветом PAPER в ивет INK, начиная с координат х. т.

Команда FILL PAPER наоборот закрашивает область экрана, окрашенную оветон IRE в пвет PAPER.

Закрашивание происходит от точки с координатами к. у во все

стороны до тек пор, пока не встретятся участки, уже окрашенные в IffIt для команд FILL IHE и

FILL иди в PAPER для КОНанДЫ FILL PAPER. Если исходная точка и ее окрестности уже окрашены в требуемый цвет, ничего не происходит.

в отличие от обычного БЕНСККа параметр пвета пос ле IЫJ или PAPER может не указываться, в этой случае идет закрашивание текущим цветом, например. FILL PAPER; х, т - сработает.

Пример:

```
10 FOB п= 1 то 6
20 CLS
30 CIRCLE 1вж п; 128. 88, п«10
вд FILL INK n; 12ft,8S
50 еехт п
```

Возможно и использование сложных конструкций FIL'.. таких как: FILL INK 2: PAPER 1; FLASH 1:ж.т

в таких случаях первое ключевое слово после FILL указывает на то. что используется при заполнении (INK или PAPER).*а остальные просто меняют атрибуты заполняемой области.

поскольку, как Вы знаете, в пределах одного знакоместа невозможно использование более двух цветов одновременно, техника работы с цветом должна быть очень хорошо продуманой. Очень легко получить неприемлемый результат, особенно в тех местах, где контактируют два овета IHZ. Обычно это обходят за счет того, что стык областей с разными цветами проводят по границам знакомест.

```
10 LET х=128. т=88, гэо--70
20 LET 8 = (SQR 2)«rad/2
30 CIRCLE X, y. rad
40 PLOT X. y. DRAW -в. -8
50 PLOT X.r: DRAW в.-в
60 PLOT x.y: DRAW 0.rad
70 FILL IHE 2; X-5. т
вО FILL IRK 4JX*5.y
```

FILL работает на областях любой геометрической формы. для примера попробуйте заполнить весь экран, за исключением того, что находится внутри букв "о". рй1ит STRIHG* (704,"о"): FILL о,о

Метод работает быстро, если есть большое количество доступной свободной памяти. Если вы заполняете большую область, например весь экран, то сможете увидеть очевидные нзуэы в те моменту, когда компьютер проверяет не забыл ли он окрасить какие-то участки, ножно реконедовать делить сложите область на несколько более простых и заполнять их порознь.

ножно установить, какое коли-чество пикселей было закрашено по последней команде FILL. Для этого

служит функция FILLEDO.

Прервать заполнение можно в любое время нажатием BREAK.

24. GET числовая переменная

или GET строковая переменная

Клавиша; G

Это то же самое, что GET от клавиатуры. Бак и INKEY* это способ чтения клавиатуры

без использования ENTER, ОТЛИЧИе от INKEY* в тон, что GET ждет нажатия клавиши. При использовании со строковой переменной, GET вводит один символ:

10 GET a$: PRINT a$; : GO то 10

Работа этой строки похожа на работу пивушей машинки. Обычный путем вы можете переключаться на печать прописными и строчными буквами. курсор можно перемешать в нужном направлении, работает стирание символов. ENTER дает переход к началу следующей строки. Более изощренная версия выглядит так: 10 GET a$ 20 PRINT CHR$ 8; " ": CHR$ в; at;

INVERSE 1; "в"; IUVERSE о; 30 GO то 10

Может быть, Вы захотите изменить размер символов, воспользовавшись командой CSIZE.

Если GET применяется с числовой переменной, то при нажатии клавиш от "1" до "9" вводится число от 1 до 9. при нажатии "а" или "а$ вводится число 10, при нажатии "в" или "ь" вводится число п и т. д. это очень удобно для организации разного рода экранных меню при написании меню-управляемых программ (см. также он).

25. GET строковая переменная, х, у <ширина, длина><;тнп>

Это как бы GET с экрана. Применяется для того, чтобы какой-то прямоугольный блок экрана присвоить некоей строкОВОЙ переменной и впоследствии печатать его на экране с помощью PRINT или PLOT в

тех позициях экрана, в каких захотите.

(Если Вы будете применять PRI8T. то необходимо учесть, что установка CSIZE не должна быть нулевой. Так, например. CSIZE в даст печать сохраненного в строкОВОЙ переменной изображения в натуральную величину. }

Размер задается в знакоместах. а координаты - в пикселах, в простейшем случае за именем строкОВОЙ переменной следуют координаты левого верхнего угла.

ю PRINT -OWE"

20 GET a$, 4. 175

30 PLOT 1оо. 100; a$

т. к. размеры снимаемого с экрана блока не указаны, то по умолчанию предполагается, что ддина к ширена равны по одному знакоместу.

в этом примере на экране будет

напечатано -OWE", а затем правая половина буквы Q и левая половина буквы W будут сохранены в переменной a$. с помощью усовершенствованной команды Бета-Бейсика PLOT Вы сможете напечатать полученное изображение в любом месте экрана (си. PLOT), причем сделать это можно с разным увеличением (см. CSIZE).

Здесь есть один нюанс, как может строковая переменная содержать часть графики экрана? Если Вам это не интересно, пропустите несколько следующих абзацев.

Поставим маленький эксперимент. Узнаем длину строковой переменной a$, это ножно сделать PRINT LEH a$. Вы увидите, что длина этой переменной равна 9 символам. Первый содержит управляющий код. который говорит 0 тон. что в последующих 8 символах идет графический образ. Если Вы сняли блок большего размера, а не одно знакоместо, то в него вставлены там, где это надо, коды управления курсором (сн. раздел Управляющие коды).

Нижеприведенный пример изображает на экране блок размером 3Х3 знакоместа, запоминает его в переменной a$ и затем печатает в случайно выбранных позициях экрана, причем делает это гораздо быстрее, чем это можно было бы сделать повторным его переетроениеи. 10 CIRCLE 1о, 165, tO 20 CIRCLE 13, 165.5 30 FILL 5. 165 40 GET a$. 0. 175, 3, 3 50 PLOT RND»23O, RND«150+20;a$ 60 GO TO 50

Если Вы используете OVER о. то увидите, что поля рисунка затрут цветом PAPER то, что лежит под ними, для получения других результатов попробуйте использовать OVER 1 или OVER 2 и запустите пример еще раз. (OVER г это режим не стандартного БЕЙСИКа. а бета-БЕЙСИКа, позволяющий Вам рисовать и без затирания и без инвертирования того, что лежит под Вашим изображением.)

если вы хотите нарисовать много окружностей одного размера, то гораздо быстрее

работают GET и PLOT, чем CIRCLE, попробуйте уда-лить строки 20 и 30 и введите OVER 2 (или используйте в строке 50 PLOT OVER 2; RND « . . . J.

Эти рассмотренные примеры относятся к нулевому типу. Если Вы не указываете при команде GET параметр <тип>. то предполагается, что он равев нулю. Нулевой тип команды GET - "бесцветшшг. То есть рисунок снимается с экрана без цветовых атрибутов и воспроизводится командой PLOT в те-куших установленных цветах иди в локальных цветах, указанных в команде PLOT иди PKINT.

Вы нохете использовать команду

GET и другого типа, это тип-1. Для его использования после ко-навды поставьте точку с запятой и поствьте 1. в этом случае изображение снимается с экрана в строкОВУ» переменную вместе с цветовыми атрибутами, точно так хе оно будет и изображено по команде PLOT или PRINT. принципиально важно, что если графический блок, с которым Вы работаете, имеет несколько цветов, то все их можно будет воспроизвести только при работе с первым типом GET. Ни глобальные, ни локальные установки цвета не повлияют. Изображение будет воепроизведено так, как было снято.

10 PRINT INK 2; -авс-

30 PRINT: PRINT IKE 4; "DEF"

30 GET s*,0, 175.3,3; 1

4Q PLOT 100.100;s*

Как обычно. вы не должны забывать, что для "Спектруиа" в пределах одного знакоместа возможны только г цвета (IRE и PAPER). поэтому тщательно планируйте, куда Вы помещаете изображение, снятое по GET вместе с цветами. при неаккуратной печати его на экране, вам не избежать проблем с атрибутами ("клэшинг" атрибутов).

26. JOIN <номер строки>

Клавиша: SHIFT * б с то же, что и "&") си. также SPLIT.

Объединяются вместе две строки, первая строка - номер которой задан (если не задан - то та строка, на которой стоит курсор) и следующая за ней строка.

Совместно используя SPLIT и JOIN Вы получаете возможность "отрезеть" часть строки и "жрис-тегнуть" ее к другой.

27. JOIH строкОВЫЙ массив или числовой массив.

клавиша: SHIFT * б

(то же. что и "&")

См, также главу "Обработка данных", с. 6.

Эта команда позволяет переместить часть символьной строки или массива в любую позицию другой

символьной строки или числового

массива. с командой JOIH тесно связана еще команда copy, поскольку они очень покожи друг на друга, то и обсуждаться здесь будут совместно. Разница их действия состоит в тон, что если команда JOIH перемешает данные, то команда COPY - копирует кк. Разница очевидна. При копировании исходный материал не уничтожается, а при перенешении он пропадает.

а) Работа с символьными

насеивали. Образец синтаксиса: JOIH a$ <п то т> то ь» <к>

COPY a$ <в то ш> то ь* <к>

Здесь:

a$ - исходная символьная строка; ь* - строка назначения; n,m - параметры выделения подстроки из строки; к - поэшшя в строке назначения.

Пример:

10 LET a$^"12345" 20 LET ь* = "ABCDEFG-30 JOIH a$ TO b*

40 PRINT ь»:

REM печатается ABCDEFG12345 50 PRINT a$:

REM: a$ не найдено.

Поскольку мы применяли JOIl и не указывали параметров п, ю, то вся переменная a$

целиком перешла в ь* и печать a$ в строке 50 ничего не даст. Если же теперь вы в строке 30 замените JOIH на COPY, то сможете убедиться в строке 50, что a$ не пострадало.

Вариант применения copy, тем самым, похож на то же, что мы инеем просто работая^ с LET:

LET b* = ь$ + a$

Но зато и JOIH и COPY работают даже тогда, когда a$ и ь* такие огромные, что заполняют всю память компьютера, а LET в этом случае работать не сможет, например LET at = a$ +• "x" не сможет работать, если a$ длиннее, чем одна треть свободной памяти компьютера.

с поиошью параметров выделения подстроки из строки Вы можете работать не только со строками, но и с подстроками. Их можно вставлять в строку назначения в задан-ное место, указав параметр позиции вставки.

Если не заданы параметры ПОДстроки, то принимается по умолчанию вся строка.

если не задан параметр позиции вставки в строку назначения, то вставка выполняется после последней позиции.

попробуйте поэкспериментировать с вышеприведенным примером, заменяя строку 30: зо JOIH a$ (E) то ь*

REM: a$-"1345', b$^"ABCDEFG2"

30 JOIH a$ (3 то) то ь$

REM: a$^"1й", b*="ABCDEFG345"

30 copy a$ то ь$(3) REM: a$;"12345-,

ь*- "ABie34CDEFG"

зо JOIH a$<гтоз) то DKLEH ъ*+п

REM: a$--145", b*="ABCDEFG23"

а теперь рассмотрим конкретный практический пример, вам надо просмотреть символьную строку t* и всякий раз. как в ней встретит-

ся слово -spectrum- заменить его на "ZX-. для этого можно использовать COPY. DELETE и IHSTRIHG. 10 LET t#="The Spectrum is a versatile computer, but Spectrum owners шаг feel it should have better editing. "

20 PRINT t*

зо LET n*="2X", o*="Spectrum"

40 LET P=I

50 LET P=IHSTRIHG (Prt*.O*)

60 IF P<>0 THEN

DELETE t*(p TO P+LEH o*-I>

COPY п* TO t*(P]

GO TO 50 70 PRINT t*

b) работа с символьными

насеивали. Образец синтаксиса:

JOIH a <п то ш> то b <k>

COPY a <n то m> то ь <к>

Здесь:

a - исходный массив; b - массив назначения; п. ш - параметры выделения

подмассива из массива; к - позиция в массиве назначения.

Массивы - это удобный метод работы с большим количеством данных, но обычно они имеют много серьезных ограничений. Наверное самое существенное из них - то. что массивы имеют Фиксированный размер, объявленный в момент назначения массива. Вы можете нерационально расходовать память, назначив Jшн) массив больше, чем Вам на самом деле необходимо, и Вам может не хватать памяти например для создания еще одного массива.

бета-бейсик позволяет более гибко манипулировать с массивами. Команды JOIH и copy позволяют перемешать иди колировать весь массив или его часть, пространство для нового материала создается внутри назначенного массива. поэтому ничего не будет перезатерто, можно обслуживать не только одномерные, но и двумерные массивы, а этого

бывает достаточно для большинства приложений, одномерные массивы трактуются, как двумерные. имеющие вторую размерность, равную единице, нет никакой необходимость, чтобы совпадали размерности исходного массива и массива назначения.

Нет также никакой необходимости, чтобы оба массива имели строго одинаковое количество рядов и их длину. Когда Вы работаете со строкОВЫМИ массивами, строки исходного массива "подрезаются" в соответствии с длиной строк массива назначения, если они длиннее или, наоборот, "подбиваются пробелами", если они короче. числовые массивы обрабатываются так же, за исключением того, что вместо пробелов используются

нули.

Предположим, что вы имеете насив a$(100.30» и он полностью запорднев. Чтобы добавить к нему еже 20 символьных строк. Вы можете действовать например так:

DIM ь*(20. 30): JOIH ь* то a$

Поскольку мы использовали JOIH. все строки будут реально удалена иэ массива ъ» и помечены в a$, а не просто скопированы, как это было бы ори применении cop?. Поскольку никаких параметров вырезки после имени исходного массива ае было указано, то все его символьные строки будут перемешены и массив перестанет существовать. при использовании COPY он остался бы нетронутым.

Позиция вставки в массив назначения не была указана. По умолчанию будет принято, что это позиция номер ю1, то есть следующая за последней. Первая символьная строка из массива ь* станет сто первой в массиве a$, а последняя станет строкой номер 120.

т.к. в бета-БЕИСНКе массивы могут часто менять свой размер, то вам может потребоваться неоднократное применение функции LEHGTH чтобы определить размер массива. в вьвеприведенном примере LEHGTH (i.-af > дало бы величину ISO.

если вы примените DIM ь*<20. 3) или Din ь* (Ю.50). а затем сделаете JOIH м то a$, то в результате строки массива ь* будут

•подбиты" пробелами или. наоборот "подрезаны" так, чтобы они укладывались в строки массива a$, имеющие 30-символьную длину.

Если у Вас уже есть готовый массив и Вы решите, что в нем надо сделать строки подлиннее. Вы можете это сделать, задав через DIM массив с желаемой длиной только с одним элементом, а потом переместить (JOIH) свой массив в него:

DIMb$(I,40): JOIH a$ то ь*

Единственный недостаток, к сожалению, состоит в тон, что теперь все наши данные будут находиться в ь», а a$ не будет существовать. Чтобы восстановить его, можно сделать обратный ход:

DIM a$(1.40): JOIH ь* то a«

Ло сих пор мы имели дело с полными массивами, но и JOIH и copy могут работать много гибче. Вы можете указать какие символьные строки или какие ряды исходного числового массива Вы хотите использовать с помощью выделяющих параметров, а также позицию, в которую произойдет вставка.

следующие примеры написаны для JOIH. но соответственно они могут работать и с COPY.

JOIH a$ то ь*(4>

- введет весь массив a$ в ь* та-

кии образом, что первая символьная строка будет вставлена после четвертой с нивольной с троки в увеличенном массиве ь*. JOIH a$<2 то 5) то ь*(1)

- переместит ч символьных строки иэ a$ в начало массива ь*. После этого массив a$ станет на четыре строки короче, чем был, а массив ь$ станет на четыре строки длиннее.

JOIH a$(10) то ь*

- переместит десятую символьную строку из at в конец ь$.

с) числовые массивы.

Одномерные и двумерные числовые массивы обрабатываются способом, похожим на символьные массивы. После имени массива должны обязательно стоять скобки, чтобы отличать массивы от простых пере-иенных-

Нижеприведенныи припер создает два массива, а затем объединяет их между собой,

ю DIM a<в)

20 FOR п=1 то S

зо LET a(пз=п

40 NEXT п

50 DIM ь<5)

60 FOR п--1 то 5

70 LET b(n)=n«10

80 NEXT п

90 JOIH ы) то a{) 100 F08 n=l TO LEHGTH (1,"a<-) 110 PRINT a(п) 120 NEXT п 130 REM печать 1 г 3 4 5 б 7 9

10 20 30 40 50 но REM ьо - не существует

28. KEYIN строковая переменная

Клавиша: SHIFT > 4

KEYIH вводит строкОВУЮ переменную так, как если бы Вы впечатали ее с клавиатуры. Таким образом, есть возможность сделать программу саносоздаюшейся. хотя это и выходит за пределы, рассматриваемые в данной инструкции. Можно рассмотреть пример автоматического создания строк DATA. 10 LET a$ = -100 DATA-20 FOR п=0 TO 9 30 LET a$ = a$ *STR* (PEEK H)

40 NEXT н

50 LET a$ = a$ (1 TO LEH a$ -1):

REM удаление последней запятой 60 KEYIH a$

После запуска данного Фрагмента Вы увидите, что к программе добавилась еще одна строка.

в режимах KEYWORDS 3 и 4 клю-чевые слова, набранные по символам, будут преобразованы в односимвольные токены.

29. KEYWORDS число.

Кдавива: в

Оператор KEYWORDS управляет тем, как вводятся и как изображаются на экране ключевые слова стандартного бейсика, бета-бей-СИКа и символов графики пользователя (UDG).

KEYWORDS о.

в этом режиме за клавишами в графическом режиме (курсор G) закреплены символы графики пользователя, как в стандартном беи-СИКе Вашего "Спектрума".

KEYWORDS 1.

в этом режиме за клавишами в графическом режиме закреплены ключевые слова бета-бейсика.

в исходном состоянии после загрузки система находится в режиме KEYWORDS 1. а если Вам надо воспользоваться символами графики пользователя, дайте команду KEYWORDS о.

Выбранный Вани режим KEYWORDS 1 или KEYWORDS о не влияет на то. как вводятся ключевые слова -по буквам или как токены. целиком, т. е. выбор режима 0 или 1 не влияет на установки режиноп KEYWORDS 2, 3 и 4. После загрузки компьютер находится в состоянии KEYWORDS 3. текущий режим ввода сохраняется вместе с вашей программой, если Вы отгружаете вместе с ней код icoDE) самого бета-бейсика.

KEYWORDS 2.

в этом режиме все ключевые слова вводятся одним нажатием клавиш, при необходимости с одновременным нажатием шифтов. Ключевые слова стандартного БЕЙСИКа берутся, как обычно. Ключевые слова бета-БЕЙСИКа вводятся в графическом режиме (курсор G), а функции бета-БЕЙСИКа вводятся в порядке FH. буква, знак "*" или " с.

KEYWORDS 3.

Этот режим - тот же, что и KEYWORDS J за исключением того, что вводимая строка, прежде чем пойти в память компьютера, сначала проверяется на наличие в ней набранных по буквам ключевых слов и. если они найдены, происходит их замена на токены ключевых

слов. по-видимому, это самый удобный режим, т. к. в нем можно работать и с вводом ключевых слов одним нажатием клавиш и со вводом их по буквам. Если нужно выйти из курсора -к", вы можете использовать ведущий пробел.

KEYWORDS 4.

в этом режиме пет курсора -к". т. е. все ключевые слова вводятся только по буквам. Форсировать появление курсора "к", тем не не-

нее. все же возможно, это делается нажатием клавиш SYMBOL SHIFT и ыггек.

этот режим, возможно, предпоч-ттт те, кону может потребоваться совместимость с другими моделями персональных компьютеров.

d) Распознавание ключевых слов.

Есть небольшое ограничение в режимах 3 и 4. которое заключается в тон, что Вы не можете использовать имена переменных и процедур, совпадающие по написанию с ключевыми словами. поскольку при запоминании программной строки в памяти компьютера может произойти их замена на токен ключевого слова. Тем не менее, ключевое слово может быть часты» имени переменной или процедуры без проблем.

Ключевые слова в этих режимах распознается как набранные прописными, так и строчными буквами. Ны Вам рекомендуем использовать строчные буквы, тогда после конверсии этих ключевых.слов в то-кена. Вы сможете наглядно увидеть, какие ключевые слова были преобразованы и. возможно, найдете синтаксическую ошибку.

Символы, стоящие непосредственно перед ключевым словон или за ним не могут быть буквами или символами подчеркивания V.

Printa - не будет конвертировано в PRINT a, т. к. компьютер предположит, что это имя переменной или процедуры.

Print a - будет преобразовано в PRINT a. Ниже приведены несколько примеров возможных строк и показано их возможное преобразование в результате "токенизации".

print 10..... PKINT ю

print forK. total .....PRINT forK.total

alter to inK3, paper1

..... ALTER TO INK 3, PAPER 1

print 3trine*<iO. "plot") ..... PRINT STRI8G*<10, 'Plot')

gOtOlO ..... GO TO 10

BO to X. ..... GO TO я

sotox..... eotox

defproc PinK. .... DEF PROC PlnK

lmat^pnnt. ..... roat.print

пример с GO то показывает, что внутренние пробелы в этот оператор можно и не включать, точно так же и "gosub", "опеггог", "defproc" будут распознаны, как полноценные ключевые слова.

слово "inK". входящее как составная часть в "PinK" и "print", входящее в "roat_print" распознаны не будут, т. к. перед ключевым словон ае должно быть буквы иди

знака "_".

30. LET переменная = число <. переменная - число...

Маленькое усовершенствование старой команды LET позволяет выполнять серию присвоений, разделяя их запятыми, это сокращает объем занятой памяти (по одному байту на каждый опушенный LET). делает ввод более удобным и листинг более читаемым, так. запись

to LET x=i.y=a»z=3,a$=-T-,b»="n" может заменить:

10 LET x=l: LET T=Z: LET 2=3: LET a$=-т": LET b*="n-

31. LIST <номер строки> то

< номер строки> или LLIST <номер строки> то

< номер строкЯ>

Вы видите по синтаксису, что здесь есть небольвое добавление к стандартному бейсику. Вы можете выводить на экран или принтер заданный Вами блок программы. если

первый номер строки не указан, то по умолчанию принимается строка, следующая за нулевой. Бели второй номер строки не указан, предполагается по умолчанию последняя строка программы. если оба параметра опушены. Вы получаете эквивалент обычной команде LIST.

LIST 20 то 1оо

LIST TO 2оо

LLIST 100 то ISO

Если строка с первым номером существует, то при листинге она будет изображена с курсором ">", т. е. она готова к вызову на редактирование.

если оба номера совпадают, то будет изображена только одна строка.

32. LIST DATA или LIST VAL или LIST VAL$

эти разновидности команды LIST позволяют распечатать сводку переменных:

LIST DATA - все переменные

LIST VAL - числовые переменные

LIST VAL» - строксвые перенен-ные.

"Спектрум" имеет б типов переменных. Команда LIST VAL распечатывает из них 4 типа в сле-дуюшеи порядке:

1. числовые массивы.
2. переменные циклов FOR-NEXT.
3. переменные с односимвольными именами.
4. Переменные с многосимвольными именами.

команда LIST VAL$ распечатает остальные два типа переменных:

5. символьные массивы.
6. Обычные символьные переменные.

Команда LIST DATA распечатает все б типов переменных.

Переменные каждого типа распечатываются в алфавитном поряжхе (для переменных с многосимвольный именем в расчет принимается только первая буква). Пример того, что может дать LIST DATA шчгаедав ниже: 4(10.4). ксз, 3.4)

n STEP I 500 LH 200 8 3. 5 J 100 3 23. 1

applen I

number 9999

хоз о

хгв 256

уоз о

Угв 176

t*(100. 10}

a$ LEH 5 "Hello"

ь* LEW 40 " Too lone to. -."

e* LEH 5 "Bangi"

Для массивов изображается только их размерность, но не содержание. Переменные циклов FOR NEXT можно отличить от прочих благодаря присутствию параметра STEP и параметра LH (looping number -нонер строки, из которой производится возврат в голову цикла). Для длинных строкОВЫХ переменных изображаются только первые 15 символов.

33. LIST DEF KEY

Здесь DEF EET располагается на

клавише SHIFT * 1.

См. также DEF кет.

Эта команда позволит распечатать список клавиш, определенных пользователем. Сначала распечатывается клавиша, а затем символьная строка или оператор, которые за ней закреплены. Если в назначении клавиши последним символом является двоеточие, то полагается, что при нажатии этой клавиши, то, что за ней закреплено, сопровождается последующим EHTKR, т. е. сразу после нажатия содержимое появляется в нижней части экрана в системном окне.

Михайленко В. С.

ЗАЩИТА ПРОГРАММ

Продолжение. (Начало см. стр. 9-16, 53-60>

Глава 4. прочие приемы защиты.

4.1 Запуск программ в кодах.

о том. как запустить программу в малинных кодах, наверное, знают все пользователи "ZX SPECTBUH". это информация изложена во всех справочниках по данному ТИПУ компьютеров.

Вкратце напомним основные положения данной системы команд.

Для вызова подпрограмм в машинных кодах используется функция USR, составленная из ключевых слов английского языка:

User SubSoutine

В компьютерах типа ZX SPECTRUM эта Функция может использоваться двояко. Во-первых, она применяется в случаях, когда необходимо вызвать подпрограмму, написанную машинными кодами и Расположенную в памяти по известному адресу.

Она также может применяться для записи данных графики, уста-навдиваеной пользователем, в зарезервированное место в конпе памяти.

Нас интересует случай применения USR для вызова подпрограмм в машинных кодах. Для запуска данной подпрограммы USR используется с ключевым словом Бейсика RAHDOMIZE или PRINT. После комбинации этих ело в ука зыва етс я цифровая величина, например:

90 RAHDOH1ZE USR зо000 100 PRINT USR 45О00

Если внесто определенного цифрового значения используется выражение, то оно должно быть заключено в скобки. Указанная величина округляется до ближайшего целого числа - это адрес памяти, с которого и должна стартовать записанная машинными кодами подпрограмма. в общем случае адрес начала подпрограммы и адрес, с которого она стартует, могут не совпадать, хотя очень часто они совпадают.

Результат функции USR - значение, находящееся в паре регистров вс микропроцессора. Комбинация ключевых слов RANDOMIZE USR иди RESTORE USR только запускает подпрограмму, тогда как FRINT USR еще и индицирует содержимое пары регистров BC на экране.

Фактически, это достаточно широко известная информация и на ней не стоило бы останавливаться.

если бы не одно но: - современные системы защиты. используя встроенные функции компьютера, создали новые системы команд для запуска подпрограмм в машинных кодах.

То, что из этого следует, очевидно. Засекретив начало вашей подпрограммы в кодах. Вы сбиваете с толку "хакеров", а это в свою очередь защищает вашу программу от несанкционированного просмотра. Рассмотрим более подробно эти новые ухищрения.

Выше были рассмотрены команды, достаточно широко известные и наиболее часто применяемые. чуть реже встречается комбинация

LET A = USR 30000

Фактически эта команда полностью аналогична рассмотренным выше

она запускает подпрограмму в кодах с адреса, указанного в функции USR - в данном случае, с адреса зо000. любопытно, что эта комбинация достаточно широко используется при работе со встроенным калькулятором "СПЕКТРУМа". (Для тех, кто интересуется этим более подробно. рекомендуем изучить трехтомник •ИНФОРКОМа" по программированию в машинных кодах, где это применение описано более подробно}.

Но, если читатель ног встретить подобную комбинацию в некоторых программах, то описанные ниже выражения встречаются достаточно редко. я имею ввиду применение для запуска подпрограмм в кодах некоторые функции Бейсика.

в частности, если Вы подадите команду:

GO TO USR ЗО000

то выполнение команды осуществится аналогично RAKDOHIZE USR ЗО000

Среди таких модификаций хо-нанды RANDOMIZE USR следует отметить еще РЯД Команд, в частности:

LIST USR и

CLOSE* USR

однако следует учитывать, что пиненение данной системы команд будет иметь весьма незначительный эффект, если не учитывать одного нюанса, а именно: несмотря на вариаоин первого ключевого слова, наличие команды запуска подпрограммы в кодах достаточно очевидно. Вот почему эту комбинацию следует использовать совместно с другими приемами.

Для пояснения вышеизложенного рассмотрим применение подобной системы команд у Billa Gilbert-a (следует отнетить, что он берет на вооружение все новые системы

защиты и поэтому изучение его методов и приемов значительно повысит Ваш профессиональный уровень. В самом деле, он взламывает программу и должен поставить на нее такую защиту, чтобы другому вскрыть ее уже не удалось. Причем следует отметить, что он использует и достаточно оригинальные методы, те свои собственные разработки, которые не встреча-ютея ни в каких других программах).

когда мы просматривали дампинг одной из программ, вскрытых биллом Гилбертом, то обнаружили достаточно любопытную комбинацию символов. После номера строки следовало:

CLOSE * USR 0 и т. д.

Сначала мы приняли ЭТУ последовательность символов за начало программы в машинных кодах, размешенной в Бейсике. В самом деле. эта последовательность не несла в себе никаких информационных признаков о командах Бейсика. Но в то же время, если бы это было начало подпрограммы в кодах, то первым должен был бы следовать символ REM. поскольку именно после него размещается подобного рода код. Это заставило нас более внимательно проанализировать данную строку и секрет был быстро разгадан.

Естественно, эта комбинация свидетельствовала о начале подпрограммы в машинных кодах, но не думайте, что она была равнозначна команде

KAHDOMIZE USR O

Ноль в данном случае был ширмой, за которой скрывалось истинное значение числа, указывающего адрес начала программы машинных кодах, этот прием с подменой значений достаточно подробно описан нами в главе 11 данного пособия.

4.2 Защита, основанная на использовании вариаций,числа PI внесто цифровых констант.

в главе I были.описаны специальные POKE которые применяются в различных целях в тон числе и для защиты от остановки командой BREAK во время работы вашей программы. Таких методов существует несколько, но общие критерии их применения аналогичны - Вам необходимо изменить значение одной из системных переменных и тогда после нажатия BREAK (если Вы ввели это изменение до нажатия данной клавиши), У вас не произойдет ос-

тановки программы, а произойдет сбой в работе компьютера и машина либо зависнет, либо произойдет сброс системы, аналогичный нажатию кнопки RESET.

но с тем, чтобы СКРЫТЬ от начинавшего "хаккера", какую именно системную переменную вы меняете и какое значение Вы туда засылаете, ножно использовать следующий прием.

В частности, вам необходимо изменить содержимое ячейки 30000, сделав его равным 150.

традиционно данная команда выглядела бы следующим образом: 10 POKE ЗО000,150

Но Ваша задача запутать "хаккера". Вместо чисел желательно использование каких-либо переменной, причем таким образом, чтобы они не описывалась в этой же программе оператором LET.

Поясним вышеизложенное на примере, чтобы заменить адрес традиционным способом, нам пришлось бы вводить еще одну строку программы

10 LET A1 - ЗО000 гО POKE A1, 150

Но этого можно избежать, если задать переменную Л, подав команду с клавиатуры:

LET A -- ЗО000

после этого значение переменной А будет занесено в память компьютера и мы сможем вызывать его по мере необходимости, одно условие - это значение будет уничтожено, если вы подадите команду RUS, т. к. этв команда полностью освежает буфер переменных. Для запуска Вашей программы необходимо использовать функцию GO то. следует отметить один нюанс - автозапуск программы по команде ыне и <со строки н) осуществляется функцией GO то н, поэтому ваше значение будет сохранено в памяти компьютера.

Другой аспект проблемы - это использование вместо числа 150 его кодового слова среди символов символьного набора компьютера. в частности, символу 150 соответствует графический код "G". следовательно, если Вы используете команды: LET A = зо000 10 POKE A, CODE "G" -(используется символ графики), то это будет полностью аналогично первой строке программы в этой статье. Кроне функции CODE можно использовать и производные функции PI:

нот PI - равноценно о SGH PI - равноценно 1 INT PI - равноценно 3

значения, которые не находят эквивалента в производных PI могут быть образованы с использованием различных комбинаций, в частности:

6 = INT PI + INT PI

в заключение этой статьи натолкнем читателя на очень любопытную мысль, в частности, раз Вашей задачей является как нохно более запутать "хаккера", то по-чену бы Вам не набрать вместо имени переменной (в нашем примере используется переменная а) имя. аналогично е одному из ключе вых слов Бейсика. Смею Вас заверить, это будет иметь поразительный эффект. в частности, постарайтесь представить себе неопытного "хаккера", увидевшего на экране следующую комбинацию:

LET USR -- зо000 10 POKE USR, SGH рг

Неправда ли. смотрится впечатляюще для тех. кто не знает, что здесь USR не ключевое слово, а имя переменной и набирается по буквам.

4.3 Ключевые слова Бейсика в "хэдере".

Одним из наиболее любопытных приенов защиты, шокирующий начинающих пользователей "СПЕКТРУМа". является использование управляющих кодов в хэдере. но не менее интересный приемом является использование в хэдере ключевых слов Бейсика. Эта тенденция просматривается во многих программам и позволяет сэкономить и без того скудное пространство хэдера с целью создания полносенного названия.

Поясним вышеизложенное на примере.

Тем, кто внимательно ознакомился с главой II данного пособия, наверняка известен тот Факт, что в хэдере содержится всего ю символов, отведенных под имя программы. при использовании управляющих кодов количество свободных символов сокращается, однако, если Вам хочется иметь полноценную заставку - заголовок, то одним из методов, позволяющий совместить это желание с желанием использования управляющих кодов, является применение в хэдере ключевых слов Бейсика.

Естественно, ключевые слова не могут в полном объеме характеризовать Ваше название и вряд ли смогут передать всю гамму названий, набираемых отдельно по буквам, но, в то же время, их применение создаст эффект высокой защищенности программы, а разного рода финты достаточно сильно действуют на непрофессионалов отталкивающим образом.

Рассмотрим любопытный пример -название программы "GAME OVER". Его можно было бы написать и по отдельной букве, но тогда это не дало бы возможности использовать в хэдере управляющие коды, которые в ноем варианте уничтожают

после загрузки на экране слово PROGRAM, для экономии неста здесь используется

оператор Бейсика OVER. Среди остальных, известных мне названий программных Файлов. приведу наиболее распространен-ные. использующие ключевые слова Бейсика: •HVS FORMAT' "HDS FORMAT"

"в. G. FORHAT" - используется биллом Гилбертом.

конечно, не все ваши названия можно выразить подобным образок, но, в то же время, следует отметить, что можно использовать слова-синонимы, аналогичные ключевым словам Бейсика, в дрбон случае, возможность такой замены Вам следует иметь в виду.

Следует отметить, что использование ключевых слов Бейсика в хэдере несколько лет назад еще использовалось для защиты программ от копирования. Когда копировщик считывал название программы, то он не мог распознать символ ключевого слова Бейсика и это вызывало сбой в его работе. Однако, последние модели копировщиков уже избавлены от этого недостатка

4.4 йерпающии заголовок.

многие из читателей наверняка пользовались программой COPY-COPY. и, вероятно, обращали внимание на тот Факт, как интересно там организован хэдер.

После загрузки хэдера на экране начинают поперемеинно нигать символы COPY-COPY, подобная комбинация не имеет принципиального значения для непосредственной защиты Бейсик-программ, но имеет косвенный эффект.

Организация защиты подобных программ воспринимается как очень хорошая, особенно среди начинающих •хаккеров". этот косвенный эффект совместно с действительно хорошей защитой, основанной на приемах, описанных в этой работе, превратят Валу программу в "неприступную" крепость для взломщиков.

Итак, вкратце рассмотрим, как работает данный, хэдер. Для тех.

в программе copy-COPY символы в заголовке имеют следующую последовательность : 16 - управление FLASH 1

255 COPY

6 - PRINT запятая го

1 - управление INVERSE

255 COPY

о

в

в данном случае набор этих символов следует рассматривать, как определенную программу, пред-

писываюшую компьютеру определенные действия. Рассмотрин, какие команда будет выполнять компьютер в данной случае.

Теоретически наш хэдер должен был бы появиться на экране (имеется в виду название программного Файла) в виде печатных символов, но в данном случае здесь нет таковых, первая пара символов 18 и

1 - это под управление FLASH и соответственно значение. устанавливающее режим нерпания. далее следует символ cop?, который рас-печатывается на экране и начинает мерцать, после этого следует уп-равляюиий код б - т. н. PRINTsa-лятая. которая перемешает курсор в следующую половину экрана, аналогично запятой, использующейся в операторе PRINT. Следующая пара символов устраивает значение INVERSE - в данном случае включает инверсный режим. Теперь нерпа-ние второго символа "cop?" будет находиться в противофазе с нерца-ниен первого, что и создает эффект попеременного мерцания.

чтобы создать такой хэдер на компьютере, Вам необходимо создать строку выгрузки в Вашей программе, в ней должны находиться операторы SAVE "название" и т. п. причем вместо управляющих ходов должны стоять пробелы, после этого Вам надо получить дамп памяти по одной из предложенных в главе

2 программ и определить местонахождение заголовка. После этого Вы засылаете в эту область памяти последовательность цифр, которая приведена выше. Естественно. приноровившись, вы сможете создавать и более оригинальные комбинации, в частности, например, поаробуете Убрать слово PROGRAM, или разместить заголовок в другой точке

экрана. Для этого Вам необходимо внимательно изучить раздел "Управляющие коды" главы II данной работы.

Поскольку подобные коррективы в хэдере не имеют принципиального значения, мы не буден останавливаться на заголовке другого цвета и в другом месте экрана (отличном от привычного). Надеемся, что читатель сам сможет сделать это.

4. 5 метод защиты, основанный на смешении программ в кодах при попытке взлона программ.

Если читатель внимательно изучил все материалы, изложенные выше, то он приобрел достаточно высокий уровень подготовки для защиты своих программ. Тем не менее, поработав с ними длительное время, он начинает осознавать, что многие из приемов, применяемых им. достаточно широко известны в сфере программистов для ZX SPECTRUM".

Возникает необходимость созда-

ния новых приемов защиты, никем до этого не применявшихся, чтобы обеспечить надежность завиты собственных программ. X этой цеди можно идти двояко: 1) создавая свои сугубо оригинальные приемы;

г) создавая новые комбинации из ухе известных приемов, улучшающие качество и надежность систем защита;

первый путь, безусловно, более надежен, но он требует виртуозного знания всех тайников компьютера, что требует, в свою очередь. достаточно целенаправленной работы с компьютером в течение длительного времени. Большинство же читателей вряд ли обладают возможностью длительно и целенаправленно изучать компьютер и поэтому мы предлагаем им избрать второй подход. Если первый подход хорош только для опытного, квалифицированного специалиста, то второй может подойти любому пользователю, внимательно изучившему данную статью. Второй метод включает в себя комбинирование изложенных в здесь приемов с целью повышения эффективности защиты, на первый взгляд выгода от такого подхода не столь очевидна, однако не стоит забывать, что порой интересная комбинация уже известных методов может иметь любопытный косвенный эффект, в качестве примера рассмотрим комбинацию из известных вам методов совмещения в Бейсик-строке машинных кодов и метода защиты от листинга с использованием управляющих кодов.

Когда мы с Вани рассматривали совмешение с Бейсиком машинных кодов, то било отнечено, что удобнее всего для Вас использовать первую строку программы для непосредственного совмещения. что связано с правильным нахождением адреса начала загрузки.

Безусловно для Вас такой метод является самым удобным, но не менее удобным он является и для взломщика. Ведь фактически в такого рода программах ему даже не приходится искать адрес начала подпрограммы в кодах - первая строка Бейсика начинается с Фиксированного адреса оперативной памяти, на который указывает системная переменная PROG и без подключенной периферии и умышленных изменений значение, находящееся в этой паре байтов указывает на адрес 23755. Соответственно, для нахождения адреса размещения такой подпрограммы достаточно к этому значению прибавить 5 (эти 5 байтов включает в себя номер и длину строки - по два байта на каждый параметр соответственно -и код оператора ЙЕН). Креме этого значение адреса начала подпрограммы нохно узнать из значения.

стоящего после RAHKOHIZE USfi иди его вариаций.

Одним из выходов является создание "плавающей" поддпрограммы в кодах и задание "фальшивого1 адреса старта данной подпрограммы.

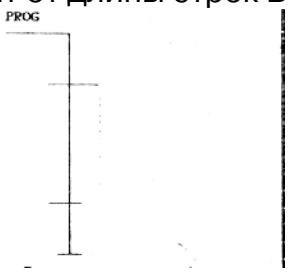
Рассмотрим более подробно каждый из методов.

как Вам ухе вероятно известно. Бейсик-интерпретатор размешает строки программ последовательно, т. е. если у Вас есть строки с номерами 5 и 10 и Вы хотите ввести строку с номером 7. то интерпретатор Бейсика разместит эту строку между строками 5 и 10. Когда он будет делать это, то строка с номером 10 переместится в более старшие адреса области озу. причем ее старое местоположение будет отличаться от нового на длину строки 7.

теперь представьте, что строка с номером ю - это строка, в которой расположена

наша подпрограмма в кодах, а строки 5 и 7 на начальном этапе отсутствуют. Та-кин образом, после Формирования данной строки с машинными кодами после оператора REM нам необходимо снестить ее в верхние адреса озу. Для этого все команды управления на Бейсике размешаем в строках, находящихся выше строки с подпрограммой в машинных кодах. (Для того, чтобы найти адрес начала подпрограммы в кодах, Вам придется использовать нетод распечатки данинига памяти, который будет подробно рассмотрен в конце этой статьи).

Если вы начнете придерживаться изложенных выше рекомендаций, то. нохно сказать, что первый недостаток нам удалось преодолеть, в саном деле, теперь у Вас нет Фиксированного адреса! с которого начиналась бы подпрограмма в кодах и этот адрес зависит от длины строк Бейсика, в которых бы осу-шествлялся запуск данной программы.



Адрес начала Бейсика. Сюда указывает системмиая переменная PROG.

В этой области находятся строки Бейсик-программ, которые осуществляют запуск подпрограмм в кодах.

Здесь находится подпрограмма в кодах.

вторая часть нашей программ» будет создаваться временно, для получения данпинга и будет располагаться после подпрограммы в кодах.

Теперь перейден ко второй час-

та проблемы. нан необходимо скрыть адрес начала подпрограммы в кодак, здесь мы используем комбинирование ухе известных методов для создания очень интересного эффекта.

в делав более полной завиты, применим двойную защиту, которая при правильном использовании превратится в тройную защиту, для этого нан придется, во-первых, использовать управляющие копы для слияния цветов INK и PAPER с ае-лью сокрытия инФорнадии, а во-вторых, создать Фальшивый адрес старта подпрограммы в кодах с использованием управляющего кода 14. Оба эти метода подробно описаны в главе г и поэтому Вам не-обходимо подробно ознакомиться с ними, ниже будет приведен лишь алгоритм создания защиты и поэтому, если Вы не уяснили себе информацию, изложенную там. Вам будет достаточно сложно сориентироваться в происходящих процессах.

Итак, для начала Вам необходимо полностью оформить аппарат Бейсика, т. е. создать строки программы, которые бы выполняли необходимую Вам операцию. Естественно, все они должны иметь номер меньший, чем строка, в которой находится подпрограмма в кодах, среди команд, помешенных в этих строках, обязательно должна быть команда запуска подпрограммы в кодах, причем номер, на который указывает RANDOMIZE USR (выгоднее применять разновидность этой команды - раздел 4.1) может быть произвольным. Обязательным условием является лишь то, чтобы он содержал не менее 5 знаков (и не более), т. е. например. запуск осушествлялся бы командой RANDOMIZE US» 85000. Это связано с тем, что при замене этого номера на истинный, который мы будем на ходить в следующей операции нашего алгоритма. программа в кодах может сместиться, из-за разного количества символов в номере.

После того, как Вы это сделали, нам необходимо будет зарезервировать место под управляющие коды. используя оператор REM. Всего у нас будет использоваться 4 байта IRK CONTROL, значение цвета IHS. PAPER сонтр и значение цвета PAPER.

для этого в первой идущей в Вашей программе строке необходимо установить REM и три произвольных символа. Оператор REM устанавливается перед первым идущим в строке оператором.

После того, как кы установим точное расположение подпрограммы в колах, мы

заменим эти зарезервированные 4 байта на управляющие коды.

следующим шагом мы как раз я
установим точное местонахождение
программы, для этого в саном
конце Бейсика разместим программу
распечатки данпа памяти.

9997 рок 1=в3755 то &4000
9996 PRIBT liTAB T; PEEK UTAB 12;
CHR реек I 9999 яехт I

Сразу же следует оговориться, что программа может останавливаться по ошибке IHVALID COLOR - неправильный пвет IUHBER T00 BIG - большое целое число и т. п. в этой случае Вам необходимо подать команду с клавиатуры: NEXT I и распечатка продолжиться.

После того, как вы стартуете данную программу командой GOTO 9997, вы получите распечатку в следующем Формате: номер десятичное символьное ячейки значение значение памяти содержащееся содержавшеея там в этой ячейке памяти

Внимательно ее анализируя. Вы более подробно изучите структуру своей Бейсик-с троки. Просматривая таким образом содержание памяти в области Вашей Бейсик-программы. Вам необходимо найти и записать номера следующих ячеек памяти: 1) Номер ячейки, где находится оператор REM. после которого следуют 4 зарезервированных под управляющие коды байта. е) Номер ячейки, в которой содержится первая цифра числа, находящегося после RANDOMIZE USR. 3) Адрес ячейки, с которой начинается ваша подпрограмма в колах,

После того, как эта информация Вани получена, необходимо занести в строку, запускающую Вашу программу и содержащую RANDOMIZE USR. правильный адрес рас положения подпрограммы в кодах. полученный в пункте з.

когда это Вами сделано, вам необходимо это число "фальсифицировать".

Как Вам уже вероятно известно, в "ZX SPECTRUM" в памяти числа хранится дважды - первый раз значение, которое печатается на экране, а второй раз - в пятибайт-яом Формате, расположенное после кода 14. в данном случае нам в? • обходимо изменить именно первое значение, которое будет распечатываться на экране.

Первый байт этого значения расположен по адресу. который кы получили в пункте г. Вы можете убедиться в этом снова, запустив программ/ дамшшга и увидев в этом же месте новое значение числа, для того, чтобы изменить это значение, Вам необходимо заслать в эти ячейки паянти значения ASCII кодов •фаяьшгаого" числа с

помощью оператора росе.

следующим пунктом нан необходимо скрыть листинг данной программы. Для этого сдедаен одинаковым цвет черяжл и бумаги с по-монью управлявших кодов. Например, мы хотим, чтобы это был белый цвет. Тогда нам необходимо последовательно заслать в ячейки памяти адреса которых мы получили в пункте 1) следующие значения: 1б - изменение IHE

7 17 - изменение PAPER

7

После того, как Вы выполните это командой роке, на экране уже невозможно будет получить лис-тинг.

Теперь Ваша программа защищена от просмотра командой LIST. Осталось удалить последние строки. осуществлявшие данпииг памяти. Сделайте это. подав команды

9997 бите»
9998 ENTER
9999 ENTER

Теперь рассмотрим, какие новые качества получила нгаа защита после такой комбинации.

Если мы вызовем первую строку программы, в которой заложены управляющие коды, для редактирования и уничтожим эти коды командой DELETE, то теоретически мы увидим адрес начала программы в кодах - практически же Вы знаете, что это значение будет

неправильным. Но, даже если Вам и удастся определить правильный адрес старта данной подпрограммы в кодах, просмотрев через PRINT реек то, что стоит после кода 14, Вам все равно не удастся ее правильно дисассемблировать. поскольку Вы уничтожили управляющие коды командой DELETE, то есть уменьшили размер оервой строки на 4 байта и, тем самым, сместили адо?с начала данной подпрограммы.

Таким образом, мы видим, что совмещение хорошо известных вам приемов может приводить к совершенно новым результатам и порождать любопытные побочные эффекты, которые в данном случае тоже используются для защиты Вашей программы.

тон 2

Техника взлома компьютерных программ ZX-SPECTRUM.

Глава 1. Введение

Компьютеры "ZX.SPECTRUM" были разработаны в Англии и поэтому многие пользователи в нашей стране сталкиваются с трудностями освоения программ. Это касается не только проблем языкового барьера. трудности, связанные со слабым званием английского языка в большинстве случаев решаются за счет его изучения самих пользователей, потону что без определенного уровня достаточно проблематично осваивать интересное программное обеспечение. Но, кроне барьера языкового, спвествует и барьер программной защита. Ведь многие пользователи, проработав определенное время с программой, желают залезть к ней в "нутро" и осу-шествить какие-либо изменения. Это могут быть изменения, направленные на русификацию программы, изменения, направленные на адаптацию программы под требования пользователя или же просто просмотр с целью изучения приемов программирования. Кроме этого. существует проблема с копированием некоторых защищенных программ, копирование которых не-возможно осуществить без определенного уровня подготовки, а также исследования конкретной структуры защиты данной программы.

в преодолении этих трудностей Вам поможет изучение техники взлома зашишенных программ, изложение которой и преследует своей целью данное пособие. Взлом компьютерных программ занятие достаточно увлекательное и, в то же время, многие используют это занятие в неблаговидных целях. в частности, очень многие взломщики оставляют после себя определенные надписи, т. е. используют взлом программ для саморекламы, но, как Вы скоро поймете, это не является самым интересным. Куда более любопытным нам представляется использование взлома компьютерных программ для повышения Вашей техники программирования, а также более детального изучения имеющегося у вас компьютера, превращения его из обыкновенной игрушки в серьезный инструмент для решения Ваших задач.

Общие рекомендации.

Если Вы внимательно изучили т. 1, то теперь Вам известны основные принципы защиты компьютерных программ, и, несмотря на то, что техника защиты постоянно совершенствуется, все, что создается отвечает известному Вам принципу: 'Все новое - это хорошо забытое старое". а это значит. что все новые приемы защиты являются модернизацией уже известных.

Но это происходит не всегда. иногда появляются исключительно новые приемы, которые нельзя классифицировать ни по какой старой схеме, к счастью, это скорее исключение, чей правило, а это

говорит о том, что вам следует рассматривать защиту новой программы как набор ухе известных Вам приемов или же их комбинацию. пока не станет ясно, что это что-то новое.

Второе, о чем Вам необходимо серьезно задуматься, так это о том. с какой целью Вы взламываете программу. Наиболее распространенным, на наш взгляд, является случай,

когда Вы желаете изучить новые приемы программирования или же адаптировать программу для себя. Эти случаи предусматривают вариант, когда вам придется досконально изучить содержание программы, чтобы понять принцип ее работы и внести необходимые изменения.

После того, как Вы определились в пели взлома, вам необходимо подумать о том. как наиболее рационально загрузить программу, чтобы наиболее быстрым способом обезвредить защиту.

для того, чтобы иметь хоть на-ло-нальское представление о типе защиты, примененном в данной программе. Вам необходимо загрузить ее и нажать клавишу BREAK ото необходимо узнать для того, чтобы впоследствии ликвидировать защиту от BREAK одним из предложенных далее способов) и если программа не "зависла", то испробовать, что получается вря нажатии клавиши LIST.

Итак, вы определились в целях взлома и осуществили предварительную подготовку. Ны котин искренне надеяться, что Ваши цели будут такими, какими представляем их мы или аналогичными.

Структура фирменной программы.

Когда Вы загружаете какую-либо игровую программу, то перед тем, как пойдет загрузка, Вы набираете с клавиатуры команду LOAD "'.

Подобная комбинация говорит о тон. что первым будет загружаться Бейсик-Файл, а уж дальнейшая загрузка пойдет в соответствии с командами данного Бейсик-Файла. Рассмотрим более подробно, как происходит загрузка этого Бейсик-Файла (Фактически данная информация относится к загрузке любого Файла в компьютере».

в течение первых 1-г секунд считывания мы видим на экране широкие красные и синие полосы, а также слышим длительный однотонный звгк. это так называемый пилот (или пилотгон. или просто

тон). который позволяет компьютеру синхронизироваться с сигналом с ленты, с которой он будет считывать программу, что обеспечивает надежность ввода и вывода кодов в память компьютера. (Как Вам уже вероятно известно, все программы состоят из кодов. Бей-

сик- программу также можно представить, как последовательность кодов), затем появляется мерцающие тонкие желтые и синие полоски, свидетельствующие о том. что компьютер считывает в память байты информации- Первый раз эти желто-синие полосы мерцают кратковременно, поскольку компьютер считывает в память заголовок. Этот э аголовок < так называемый "хэдер") и содержит в себе всю информацию о последующем загружающемся блоке кодов и имеет размер всего 17 байтов. Поскольку у нас считывается Бейсик-Файл, то появляется надпись "PROGRAM", но возможно появление и других надписей, в частности:

BYTES - байты (коды);

CHARACTER A RRAY - символьный массив;

HUHBER A RRA7 - числовой массив.

После этой надписи будет стоять название программы.

Теперь после небольшого перерыва начнется второй пилотгон, а после него считывается собственно сана программа).

Если мы подавали команду LOAD "" . то у нас загружался Бейсик-файл и после загрузки компьютер поступает в соответствии с данными, поступившими из хэдера. Для Бейсик-программы это может быть либо автостарт программы со строки п (если программа была записана на магнитофон командой:

SAVE "название • LINE либо остановка в ожидании команды RUH.

в большинстве Фирменных программ происходит автостарт и, в соответствии с командами Бейсика, выполнение программы продолжается, что позволяет осуществить дальнейшую загрузку блоков. Но теперь вам достаточно очевидно, что если взломать Бейсик-файл и внести в него соответствующие изменения, то Вы сможете самостоятельно и целенаправленно отслеживать процесс загрузки и управлять им.

Именно Бейсик-файл наиболее серьезно защищается производителями и именно на нем Вам следует акцентировать свое внимание. поскольку именно тан содержится первичная информация о процессе загрузки последующих блоков, изменив которую Вы сможете воздействовать на дальнейший провесе загрузки.

Естественно, следующим этапом нашим совместных исследований должно стать рассмотрение приемов вскрытия данного Бейсик-Файла.

небольшая историческая справка.

поскольку данный раздел называется "Структура фирменных программ", мы постараемся вам изло-

жить в обших чертах эту структуру, несмотря на то. что могли только ограничиться информацией о структуре, необходимой для взлома,

итак, как вы ухе теперь поняли, загружающийся в компьютер файл состоит из двух частей: хз-дера и собственно загружающейся после него программы (для удобства будем впоследствии называть эту совокупность блоком кодов). в хэдере содержится вся информация о загружавшемся после него блоке.

Таким образом. структуру первых прогреми для "ZX SPECTRUM" можно представить, как совокупность Бейсика и машинных кодов. Нашинные коды составляли сану программу, в то время как Бейсик осуществлял загрузку этих кодов в память компьютера, после чего осуществлялся их запуск из Бейсика с использованием функции USR. Таким образом, Бейсик состоял из совокупности команд LOAD " CODE которую завершала Функция USR (Автор не включает сюда все разнообразие текстовой информации, которая выводилась на экран с использованием оператора PRINT». Взлом таких программ с целью изучения вообше являлся личным делом. в последовательность команд Бейсика вводится оператор STOP между совокупностью команд LOAD •" CODE и функцией US8. после чего программа в необходимой для Вас несте останавливалась и Вы могли беспрепятственно исследовать как структуру, так и содержание самой программы. Фактически мы можем сказать, что эти программы не были зашишены. что трудно сказать о более поздних Фирменных разработках. Несмотря на все разнообразие появившихся типов защит, мы можем выделить два направления защиты в области загрузки. Это загрузка с хэдерон и загрузка без хздера.

Как действует спектрумовская загрузка с хэдерон, нам уже известно. был. правда, вариант завиты с нестандартным хэдером, который отрабатывался специальной программой. Для загрузки программ этот метод не нашел широкого применения и был вскоре практически полностью вытеснен методом безхэ-дерной загрузки, однако впоследствии программисты вновь обратились к нему, когда для "Спектру-на" вышла масса программ с догру-жаюнинис я уровнями, именно при догрузке уровней этот метод получил наибольшее распространение.

второй метод - т. н. загрузка без хэдера применяется практически во всех Фирменных программах последних годов выпуска. V него сувествует множество разновидностей, но Фактически все они сводятся к созданию новой системы хранения данных о вводящимся в компьютер блоке кодов, то есть

Фактически, ту информацию, которую раньве передавал в компьютер хэдер, теперь передает специальная программа в кодах. Разновидностями же этого метода безхэлер-ной защиты являются обращение либо ко встроенной в ПЗУ подпрограммы загрузки, либо обращение к одной из загружаемых в ходе работы программы подпрограмме, так называемой программе-загрузчику последующих блоков. Следует отметить, что применение своей программы-загрузчика преследует несколько целей:

- создание необычных эффектов в ходе загрузки;

перемещение загруженного блока кодов в другое место памяти, чтобы обеспечить защиту;

- выполнение этих функций одновременно.

наглядным примером первого типа являются программы MIKKI и EUHG FU второго типа GREEH венет и третьего - вонв JACK.

Разумеется, здесь перечислены не все пели, ради которых применяется конкретная программа-за-грузчик собственного изготовления. Но в обшей случае применив такой

программы говорит о методе бесхэдерной загрузки.

Применение загрузки без хэдера становится возможным благодаря знанию встроенных процедур "ZX-SPECTRUM". в частности, чтобы вызвать встроенную подпрограмму загрузки. Вам необходимо внести в регистровые пары HL и BC соответственно адрес, с которого начинается загрузка кодов и их длину, а также соответствующим образом изменить содержимое аккумулятора, после чего вызвать подпрограмму в кодах.

(очень подробно встроенные процедуры описаны в книге IAH LOGAH, FR. o'нака "THE COMPLETE SPECTRUM ROH DISASSEMBLY". 1983, а также информацию о процедурах загрузки можно получить в zx-ревю н5. б 1991г.)

Следует отметить, кроне всего вышеизложенного, что достаточно часто применяется комбинированная загрузка, когда наряду с блоками с хэдерон загружаются блоки без хэдера.

для того, чтобы читателю получить достаточно полное представление о структуре своей программы, рекомендуем загрузить ее в копировщик, который достаточно наглядно покажет Вам из каких блоков состоит ваша программа.

в копировщике ZX COPY 87, TF cop? и им аналогичным хэдер можно отличить достаточно легко потому, как именно в нем содержится информация о том, что это либо Бейсик, либо коды и т. д. . а также сано название Файла. Разумеется! в копировщике вы можете подробно изучить лишь стандартный хэдер. в то время как нестандартный подобному исследованию не поддается. Структура хэдера.

Итак, теперь мы с Вами в общих чертах рассмотрели структуру Фирменной программы. Конечно, каждая конкретная программа имеет свою структуру (в этом заключается ее оригинальность). но практически каждую из них можно классифицировать по способу загрузки - либо загружается файл с яэлерон, либо загружающийся блок не имеет хэдера.

(любопытно, что несколько лет назад появился оригинальный метод защиты программ от копирования. Он заключался в тон. что в название программы наряду с обычными символами включались и ключевые слова Бейсика, либо управляющие коды. После загрузки этого "Фадьшхэдера" в копировщик. копировщик не мог распечатать эти символы и работа программы останавливалась с сообщением об ошибке. великолепный комплект польских копировщиков ZX copy 67, TF сорт. TF COPY г, гае COPY и др. практически полностью с вел на нет все усилия в данной области защиты, интересным напоминанием этого направления защиты от копирования может служить программа GREEK BE RET, один из ФаЯ-лов которой длиной 17 байт как раз и должен был выполнять роль фальшхэдера.

Следует еще коротко сказать о том. как создатели копировальных программ вышли из положения, в последних разработках копировщиков вместо символов, не соответствующих ASCII кодам какой-либо литеры печатается вопросительный знак или иной заранее принятый символ. *

однако, немс отря на то, что метод, казалось бы. исчерпал себя, подробное его изучение может натолкнуть читателя на создание новых оригинальных разработок, базирующихся на этом или аналогичном принципах).

Теперь на предлагаем вам ознакомиться со структурой хэдера. Сделать это необходимо не только из чистого любопытства. Ведь невозможно создать что-то новое, не изучив досконально действующий старый образец.

длина заголовка 19 байтов, а не 17, как написано в большинстве книг, но только 17 должны быть активны. поскольку 'LOAD* (эта информация аналогична и для -SAVE-) первый и последний байты определяют сами. Первый байт для заголовка всегда равен нулю, а последний - "байт четности" генерируется стандартной процедурой SAVE и потому нас не интересует.

Байт г содержит число, характеризующее тип записи, в зависимости от того, какое здесь содержится значение, компьютер определяет структуру Файла, следу-вяего за хэлерон:

0 - это Бейсик-программа

1 - числовой массив

2 - массив символов

3 - блок кодов

Байты 3-1г содержат в себе имя программы

Байты 13 и 11 - длина основного блока (например для Бейсик программы это разность того, что содержится в системных переменных

ELIHE и FEOG.

Байты 15 и 16 - начальный адрес загрузки кодов или номер строки автостарта для программ, написанных на Бейсике.

Байт 16 - для массива данных здесь располагается его имя в следукней Форме: биты 0-4 - имя (от a^1 до 2=36) бит 5 - сброшен, если массив

числовой бит б - активен, если массив

строкОВЫЙ

бит 7 - активен всегда

Байт 17 и 13 - длина Бейсик-программы, т. е. разность того, что содержится в системных переменных VA8S и PROG.

Байт 19 - активизвруется в коде работы программ "LOAD" и -SAVE".

Для наглядности приведен схему заголовка (см. рис. 1):

Поскольку хэдер анализируется интерпретатором с помощыэ индексной адресации, то внизу каждый байт дан как с иешение относ и -тельно базового адреса, содержащегося в IX.

Небольшой комментарий по поводу некоторых байтов хэдера. в зависимости от считываемого блока байты 15 и 1б интерпретируются по-разному. в заголовках программ, написанных на Бейсике, эти байты содержат номер строки, с которой запускается программа, если она была записана с поновью оператора

SAVE "имя" LISE в

Если программа была записана без оппин LIHE и после считывания не запускается автоматически, то значение этого числа больше

32767. ЛОбОПЫТНО. что одним из

способов нейтрализации саносамо-запускаюонхся программ на Бейсике является замена этих двух байтов на число большее 38767 (Подробно этот метод будет рассмотрен в одной из статей Главы г под названием "Блокировка автозапуска").

Байты 17 и 18 содержат число, указывавшее длину самой программы на Бейсике, т. е. содержимое памяти от байта, указанного системной переменной PROG до байта, определяемого системной переменной VARS. Если мы от всей длины блока (байты 13 и 14) отнимем это число, то гзнаен сколько байтов в этой программе занимают вереиен-вые Бейсика.

Байты 1 J3. . . 1й 13. 14 15, 16 17, 1в 19						
Флаг	тип	Ви я	длина	Старт	Длина для Бейсика	четность
ix*. .. 0 1. ., 10 11, 13 13, 14 15. 16 17						

Рис. 1 Структура "хздера"

теперь Вам достаточно хорово известна структура вэдера и Бы уже представляете, чем является эта совокупность байтов. Однако, для того, чтобы исследовать хэдер более подробно, я рекомендую Вам использовать специальные программы, которые прозондируют загружавшийся хэдер и выдадут информацию о ней

Ниже в качестве примера приведена такая программа, фактически она написана на Бейсике, но в своей работе обращается к процедурам в машинном коде, программа загружает хэдер в определенное место памяти (которое, кстати, вы можете изменить), после чего Бейсик-программа анализирует структур? хэдера и выдает всю информацию о ней на экран в виде письменных сообщения.

Сначала программа Формирует процедуру в машинных кодах.

используя бЛОК DATA строки 30.

Фактически этот блок осуществляет запуск встроенной в ПЗУ подпрограммы-загрузчика, но делает это так. чтобы загружался только хэдер.

при загрузке хэдера в аккумуляторе процессора должен содержаться о, а в регистре IX адрес, с которого начинает грузиться хэдер. кроме этого, должен быть включен Флаг переноса (Флаг с регистра F).

```
1 вен таре EXAMINER 5 CLS:BEEP .1.24:
PRINT 'LOAD A HEADER AMD WAIT.
PLEASE":
PAUSE 150:BEEP . 1,24:CLS 7 RESTORE 10 CLEAR 32511 20 FOR A=32512 TO 32521: READ B: POKE
      A. B: NEXT A 30 DATA 175.55,221.33,16.127.
205. 86. 5.201 40 LET B=32528:
DEF FH A{X)-PEHC(B»X>«
256"PEEI(B«X«II 50 RAKDOHIZE USR 32512 60 LET C=PEEK в 70 IF C>3 THEN GO TO 50 75 LET
      B»="-:
FOR A=B*1 TO B'10:
LET B»=B»>CHR$ PEEK A: NEXT A
78 FOB A=10 TO 1 STEP -1
79 IF B»(A>=" •
THEN LBT B» = B«(TO A-l): •EXT A
SO IF B»=" • THEN LET B»:- " 85 PKINT "FILE HAKE: "i
INVERSE I; B> 100 PRINT "FILE type: • 110 GO SOB 1000»100»C 120 PRIIT :ря1ят 130 готе в.
      255 140 GO TO 50 150 STOP
160 SAVE "TAPEXAH" LIKE 5: CLS:
PRINT "O. J, REWIHD AMD PLAY": VERIFY "TAPEXAH": CLS:
PRINT "O. K. VERYFIED": STOP 1000 PRirr "PBOGRAH" 1010 PRINT "TOTAL LEHGTH: ":
FH At 11); "BYTES-1020 PRINT "PROGRAM LEHGTH: ":
FH A(15);-BYTES-1030 IF FH A<13»9999
THEN PRINT "SAVED BY:" •TAB 5; "SAVE " " •: B»: •••-:-          RETURN 1040 PRINT "SAVED
      BY:"
•TAB 5; "SAVE " " -;B»; • " ' LIHE ";FH AI13) 1050              RETURN
1100 PRINT "HUHBER A                          RRAY-1110 LET Ai=" ":
GOTO 1220
1200 PRINT "CHARACTER A                       RRAY-1210 LET a$-"»" 1220 PRINT "A
                          RRAY LEHGTH: "
:FH AI11); "BYTES" 1230 LET D=PEEt <B«14> 1240 PRINT "ORIGINAL A
             RRAY HAME:" :CHR»(64»32MD/32-INT(D/32>>> :A>
1250         RETURN 1300 IF FN A(ll)-6912
             AND FS A(13)=16384
THEN PRINT -BYTES-SCREEH»":           RETUR»
1310 PRINT "BYTES-1320 PRINT -START ADDRESS: "
;FH A(13) 1330 PRINT "LEHGTH: "
;FH Alll); -BYTES-1340          RETURN 1350 CLEAR:
SAVE -тарехаи- LIHE 5: BEEP . 1.24:
PRIBT AT 10,o: -o. E. REWIHD анг PLAY TO VERIFY •: VERIFY "TAPEXAH": CLS:
BEEP. 1.24: STOP
```

Листинг ассемблера программы в кодах, которая содержится в строке 30 пата.

7F00 хок л Обнуление
аккумулятора.

7F01 SCF принудительное включение флага переноса.

7F02 LD IX, 7F10 в Регистр IX
поместили адрес с которого начнется размещение хэдера в памяти компьютера.

7F06 CALL 0556 вызов ЗЗГРУЖа-юоей процедуры пзу.

7F09 RET

Глава 2. Блокировка автозапуска.

Введение.

после внимательного ознакомления со структурой Фирменных программ- читателю становится очевидно, что для взлома программы необходимо, прежде всего, исследовать Бейсик-файл (первичный загрузчик), этот вывод стал очевидным и для фирм-производителей программного обеспечения. Вот почему многие методы запиты

направлены в основном на защиту Бейсик-файл а.

Однако, следует заметить, что наряду с техникой программирования совершенствовалась и техника взлома, в этой области существует очень мало информации, но мой вывод подтверждает тот факт, что вскрытие хаккерани программ можно найти среди игр самых разнообразных подов выпуска.

в этой главе рассмотрены три концепции блокировки автозапуска программ. Естественно, каждую их этих концепций можно использовать независимо от других. Можно, однако применить и комплексный подход. Но основным является то, что каждый их предложенных вашему вниманию подходов имеет свои достоинства, но он в то же время имеет и свои недостатки.

Мы думаем, что ознакомившись с этими концепциями более подробно. Вы выберете для себя ту, которая наилучшим образом удовлетворяет Вашим потребностям.

2. i Загрузка Бейсика через блок кодов.

После того, как читатель озна-1комился со структурой хздера. ему становится ясно, что Фактически между заголовком Бейсика и заголовком кодов разница небольшая. а разницы между непосредственным файлом Бейсика и непосредственным файлом кодов нет вообше никакой.

Примечание: в данном случае под файлом подразумевается та часть программы, которая загружается после хэдера. на самом деле отличия есть и заключаются они в следующем:

- Бейсик-файл загружается в память компьютера начиная с адреса, задаваемого системной переменной PKOG. блок кодов же загру-

жается с адреса, указанного в хэдере.

- Бейсик-файл обрабатывается интерпретатором как определенная последовательность символе в-кодов Бейсика. Блок кодов обрабатывается процессором как последовательность кодов ZBO.

на этом свойстве этих Файлов и основан данный метод взлома, в саном деле, нам ничего не стоит обмануть компьютер - он думает, что загружает в память файл кодов, а на самом деде загружает файл Бейсика и при этой он не сможет установить никакой ошибки. поскольку при той проверке, которую осуществляет "ZX SPECTRUM". такую ошибку установить просто-напросто невозможно.

Итак, рассмотрим этот метод более подробно.

нам необходимо изучить структуру Бейсик-Файла. при этом, чтобы избежать запуска команд и процедур, осуществляющих защиту, нам необходимо загрузить программу в память без автозапуска, т. е. так, чтобы она не стартовала, если в ней предусмотрена такая возможность.

Для того, чтобы решить данную задачу, нам необходимо прежде всего узнать длину Бейсик-файла. Это можно сделать, воспользовавшись вышеприведенной программой для анализа хэдера. То же можно сделать и воспользовавшись программой -коеировшиком.

Обычно в копировщике мы можем наблюдать следующую картину (рис. г):

После имени программы обычно появляется сообщение о типе данной программы: PROGRAM, CODE и т. п. строка автозапуска появляется только для программ Бейсика (для кодов же появляется номер ячейки памяти, с которого начинается загрузка). в разделе

•длина" появляется длина загружаемого вслед за жэдерон файла. Это не длина хэдера.

нам в нашей работе понадобится

•реальная длина" (в большинстве случаев это значение не совпадает со значением "длина" в хздере. но бывают и исключения). Кроне этого Вам необходимо запомнить номер строки автозапуска.

Следующий этапом нашей работы является создание нового хэдера, благодаря которому мы сможем загрузить нашу ПРОграмму в произвольное место памяти, как Вам уже вероятно известно, компьютер не в состоянии отличить файл Бейсика от файла кодов, этим мы с вами и воспользуемся, мы создадим хэдер. который якобы должен загружать коды с определенного места памяти, но вместо Файла кодов мы за-грузин туда файл Бейсика для

последующего изучения.

Для создания такого хэдера нам понадобится значение "реальной длины" Файла, взятое из кодировщика. Теперь заштаен на ленту этот хэдер кодов командой с клавиатуры: SAVE "иня Файла$ CODE зо000, реальная длина

в графе "имя Файла" Вам необходимо будет написать произвольное название Файла по Вашему выбору, но не более 10 символов, значение зо000 после CODE означает, что файл кодов, загружающийся после этого хэдера. будет загружаться с ячейки памяти зо000 (это же касается и записи информации т. е. после подачи данной команды, на ленту, должен будет записаться блок длиной 'реальная длина$, начиная с ячейки памяти зо000. Однако, поскольку нам необходим только хэдер, мы не будем дожидаться выгрузки на ленту Файла кодов) и выключим магнитофон.

Теперь мы имеем следующие магнитофонные блоки:

Исходный файл Спеииаль-

хэдер (1) Бейсика ный хэдер

Бейсика (!') кодов (2)

этот хэдер этот файл этот хэ-мы имели описывал- дер мы вместе с ся исход- создали блоком нын кеде- таким об-Бейсик рои и за- разом. программы. грузить в чтобы он длина хэ- компьютер описывал дера 19 автозапус- файл Бей-байт, ком. сика как файл кодов. Длинам 9 байт

Теперь нам необходимо непосредственно осуществить подмену, для этого подадим команду с магнитофона:

LOAD "" CODE

после которой загрузим созданный нами хэдер. после этого хэдера вам необходимо загрузить файл Бейсика {!'), однако этому файлу не должен предшествовать исходный хэдер (п. (в данном случае этот исходный хедер <1) не загружается вообше). Наилучшее техническое решение, позволяющее точно найти файл Бейсика (!'), пропустив исходный хэдер, является временное выдергивание шнура загрузки из магнитофона совместно с использованием клавиши пауза магнитофона (если такая имеется).

Итак. Вы осуществили подмену и загрузили Бейсик-файл под видом кодов. Теперь Вашей задачей является просмотреть этот файл и изучить его структуру.

(Продолжение следует.)

иня программы	тип программы	строка автозапуска	Длина
хххххххххях	р	10	423
тутуууутттт	с	. 39000	гггъо
zzzz2zzzz2z	с	6i200	1800

Рис. г

Дж, Хардман„ э.Хьюзоы перевод Тихоновой: н. а.

40 ЛУЧШИХ ПРОЦЕДУР

Продолжение.
начало сн. с. 17-28, 61-70.

7. ПРОЦЕДУРЫ ОБРАБОТКИ ПРОГРАММ.

7.1 Удаление блока программы.

Длина: 42
Количество переменных: 2
контрольная сумма: 5977
Назначение: Эта программа удаляет блок BASIC-npor-раммы, находящийся между строками, опре-деленныни пользователем.
Переменные:
Имя - start line no
Длина - г
Адрес - 23296
Комментарии: Номер первой строки. подлежащей удалению.
Имя - end line no
Длина - 2
Адрес - 23298
Комментарии: Номер последней строки, подлежащей удаление.
Вызов программы:
RANDOMIZE USR адрес
контроль ошибок:
Если имеют место следующие
ошибки, то процедура останавливается без удаления строк BASIC-
програмиы:
- если последний номер строки меньше, чем первый номер строки:
- если между этими двумя строками нет программы на БЕНСИКе;
- если один из номеров строк или
оба равны о. Комментарий:
Эта программа довольно медленна для удаления большого блока программных строк, но. тем не менее, работать с ее помошью все же удобнее, чей удалять строки вручную.
не вводите номера строк больше. чем 9999.

```
ЛИСТИНГ НАЯИННЫХ КОДОВ МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,(23396) 42 0 91
            LD DE,(23298) 237 91 2 91
            LD A,H 184
Ой L 181
            RET Z 200
            LD A,D 122
            ORБ 179
            RET Z 200
            PUSH DE 213
            CALL 6510 205 110 25
            EX (SP).HL 227
            INC HL 35
            CALL 6510 205 110 25
            POP DE 209
            AND A 167
            SBC HL, DE 237 88
            RET Z 200
            RET C 216
            EX DE,HL 235
```

```
SXT.CH:          LD A,D 122
            ORE 179
            RET Z 200
            PUSH DE 213
            PUSH HL 229
            CALL 4120 205 24 16
            POP HL 225
            POP DE 209
            DEC DE 27
            JR HXT.CH 24 243
```

Как она работает:

в пары регистров HL и DE загружаются начальный и конечный номера строк соответственно. Эти значения проверяются и. если одно из них или оба равны о. программа возвращается в BASIC.

Затем вызывается подпрограмма ПЗУ по адресу ь510 - она возвращает адрес в памяти компьютера, с которого начинается первая строка. Эта хе подпрограмма затем вызывается снова для определения адреса символа, стоящего после 'ENTER' в конечной строке.

в пэру регистров HL помешается разность этих двух адресов и, если это значение равно 0 или отрицательно, программа возвращается в BASIC.

Содержимое пары регистров HL копируется в DE для использования DE в качестве счетчика. Если счетчик равен о, то работа процедуры завершается заканчивается, а если нет, то вызывается подпрограмма пзу. расположенная по адресу 4120, которая удаляет один символ. После этого - возврат к 'нхт.сн'.

7. 2 Обмен токена.

Примечание: под "токенон" подразумевается любое ключевое слово (команда, функция) из "словаря" БЕЙСИКа. которое рассматривается данной программой (как и интерпретатором компьютера) в виде определенного кода. Длина: 46

Количество переменных: 2 Контрольная сумма: 5000 Назначение:

меняет любое вхождение заданного токена в бейсик-программе

на другой токен (например, все операторы PRINT могут быть изменены на LPRIRT). Переменные: имя - cbr old Длина - t Адрес - 23296

Комментарий: Код заменяемого токена.

Имя - cbr new Длина - 1 Адрес - 23297

Комментарий: код замеваювего токена.

Вызов программы:

RANDOMIZE USR адрес Контроль ошибок:

Если бейсик-программы нет в памяти или один из двук заданных токенов имеет код меньше 32, то процедура возвращаетеи в бейсик.

Комментарий:

Эта процедура очень быстра, но чем длиннее бейсик-программа, тем медленнее она работает.

листинг малинных кодов

```
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD BC,(23296) 237 75 0 91
            LD A.31 62 31
                        CP B 164
            RET HC 20S
                        CP C 185
            RET HC 208
            LD HL,(23635) 42 83 92
HXT_CH            INC HL 35
            INC HL 35
JHC HL 35
CHECK       LD DE, (83627) 237 91 75 92
            AND A 167
            SBC HL, DE 237 82
```

```
            RET NC 208
            ADD HL, DE 25
            INC HL 35
            LD A, (HL) 126
            INC HL 35
                                    CP 13 254 13
            JR Z, HXT_CH 40 237
                                    CP 14 254 14
            JR HZ.COHP 32 3
            INC HL 35
            JR NEXT CHR 24 330
COHP        DEC HL 43
                                    CP C 185
            JR HZ. CHECK 32 229
            LD (HL).B 112
            JR CHECK 24 226
```

как она работает:

в регистры Вис загружаются новый и старый токены соответственно. Если любой из токенов имеет код меньше, чем 32, то программа возвращается в BASIC.

в пару HL заносится адрес на-

чала БЕЙСИК-программв. пара регистров затеи увеличивается и сравнивается с адресом области дрренонди. Если HL не невъше, •ямм адрес вякала области пезаенен <авв, программа возвращается в

исшд'и

вая на сдедлжй символ. Код это-по -мит in загружается в аккгму-дятор, и HL уведагашается вновь. если зятева^ в аккумуляторе равно 13 ней 14 (ентек или нинвЕвл иодпрограниа возвращается к 'нхт_сн', , а HL увеличивается, указывая на сжежуюший символ. Если аккумулятор не содержит 13 или 14. крааямое значение сравнивается с 'cbr old'. Если пара найдена, этот синвол запевается на 'СЬг new'.

' Затеи - возврат к проверке ('свесе') аа конеп обрабатываемой программы.

7.3 Удаление REM-строк.

Длина: 1)?

Количество переменных: о

контрольная с^ниа: 13809

назначение:

Эта програина удаляет все комментарии из Бвисик-орограюш. Вызов программы:

RAHDQHIZE us» адрес-Контроль ошибок:

Если в памяти нет беисив-программы, процедура вернется в ии-снк беэ каких-либо деистам). Комментарий:

Процедура' JI3V. которая используется для удаления, символов, выполняется не. очень быстро и поэтому процедура "Удаление REM" работает медленно.

ЛИСТИНГ НАМИННЫХ КОДОВ ИЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА

```
            LD HL, (23635) 42 S3 92
            JR CHECK 24 31
NEXT_L POSH HL 229
I8C HL 35
J8C HL 35
            LD C, (HL) 78
            INC HL 35
            LD B, (HL) 70
ЙХТ-СН            INC HL 35
            LD A, (HL) 126
                                    CP 33 254 33
            JR C, HXT_CH 56 250
                                    CP 234 254 234
            JR HZ,SEARCH 38 26
            INC BC 3
            INC BC 3
            INC DC 3
```

```
            INC BC 3
            POP HL 225
DEL_L       PUSH BC 197
            CALL 41EO 205 24 16
            POP BC 193
            DEC BC 11
LDA.B 120
            ORC 177
            JR HZ,DEL_L 32 246
CHflCK            LD BE. (23ЙВ7) 237 91 75 92
            AND A 167
            SBC HL, DE 237 82
KT NO 206
            ADD HL, DE 25
JP NEXT_L M 244
SEARCH            INC 'ffi. 35
            LD A, IHL! i«6
OF 13 КЯЛ 13
Jfi ;«Z,HJEMT 5Й 8
ЯмТЛ>       POP HL 2B5
ABD HL,BC 9
            INC HL 35
            INC KL 55
            INC HL 95
IMC HL 35
JS CHECK 24 231
H_K8T                                 CP 14 2W 14
JE HZ, H.HUHB 3J 7
            INC HL 35
            INC HL 35
            INC HL 35
            INC HL 3J
            INC HL &
            JR SEARCH 24 231
N НШВ                                 CP 33 254 33
J8 C, SEARCH 5^ 227
                                      CP 34 254 34
            JR HZ,H_eUOT 32 Й
FD_QT       INC HL ' 35
            LD A,XHL) 126
                                      CP 34 ' 2Й '34
            JR HZ;FD_»T э> aso
            JR SEARCH 24 ?i5
H^GOOT                                      CP 5fl 2S4 58
            JR HZ. SEARCH Ja 211
            LD D.-H M
            LD E.L 93
F»_EHT            INC HL 35
            LD A, (HL) 126
                                      CP 13 254 13
            JR Z.E»T_FD 40, 209
                                      CP 35 S54 33
            JR C,FD_BHT. 56 246
                                      CP 234 254 234
J8 HZ,H_QUOT 32 236
            LD H.D 98
            LD L..E 107
DEL.CH            PUSH BC 197
            CALL 4120 205 24 !6
            POP BC 193
            DEC BC 11
            LD A, (HL) 126
                                      CP 13 Й54 13
            JR HZ.DEL_CH 32 245
            POP HL 225
```

```
            INC HL 35
            INC HL 35
            LD (HL),C 113
            INC HL 35
            LD <HL),B 112
            DEC HL 43
            DEC HL 43
            DEC HL 43
J» CHECK 24 160
```

Как она работает:

В пару регистров HL загружается адрес начала бейсик-области и делается переход к подврограмие проверки кокда обрабатываемое программы. -.Если конеп достигнут,

происходит возврат в бейсик.

Программа переходит к обработке следлсвея строки "NEXT_L'. Адрес, имевшейся в HL сохраняется на стеке для дальнейшего использования, а в BC загружается длина обрабатываемой бейсик-с троки. Подпрограмма •нхт.сн* увеличивает адрес в HL и загружает в акку-кпят01» очередной символ. Если его код тмишь, чем 33; т. е. это пробел - кяи управляювив символ, пронелэ^а ъо^вранается, чтобы повторить эту'' часть вновь. Если встретннйнйся символ не является сянвояоМ-'кен, делается переход к 'SEARCH',

Если jraepa-rop REM найден, ре-гйстр ас увеличивается на 4 (он может бить теперь использован в качестве; счетчика), а HL восстанавливается из стека. Затеи удаляется количество символов, определенное в вс, начиная с адреса, определенного в HL, для удаления используется пропедура цзу, расдоложенная по адресу 4120. Затем цродедура снова переходит л.продедуре 'CHECK'.

Если. происходит переход к процедуре ;SEARCH', HL увеличивается, указывая, на следуюяий символ, и это значение загружается в аккумулятор. Если это символ .ентек, т. е. достигнут конеп строки, то HL восстанавливается Из стека и увеличивается, указывая на начало следующей строки, а затбн происходит переход к 'снвсе'.

Если аккумулятор хранит символ HUHBER (код равен 141. то HL увеличивается, указывая па первый символ после обнаруженного числа. и процесс поиска повторяется.

Затем делается проверка для символов, код которых неньве 33. Если такой символ обнаружен, делается возврат к 'SEARCH'.

Если обнаружен символ "кавычка" (34). процедура выполняет никл до тек пор, пока не будет обнаружена вторая кавычка, а затем поиск будет продолжен. если найденный символ не двоеточие, определяющее строку с несколькими выражениями, поиск повторить.

HL копируется в DE, чтобы сохранить адрес двоеточия, а затем HL увеличивается, указывая на следующий символ. Если это символ ENTER. делается переход к 'EHT_FD', иначе, если это упрзв-лямий символ или пробел, программа возвращается к 'FD_EHT'.

Если символ не является кодом йен. происходит переход к 'H^QUOT'. Если же код REM найден, в HL загружается адрес последнего ветретиниегося двоеточия, а затем все символы от адреса, находящегося в HL до следующего символа ентек удаляются. Указатели для этой строки корректируются. HL устанавливается в начало этой

строки, и происходят переход к 'снес!'.

7. 4 Создание REM-строк.

Длина: 85

Количество переменных: 3

Контрольная сумма: 9526

Назначение;

Эта проиедура создает выраже-нве RE4 с заданным количеством символов в заданной строке, символ выбирается пользователей. Переменные; имя - une number Длина - г Адрес - г329Б

Комментарий: Номер строки, в которую надо вставить выражение 8ен. имя - number chr Длина - г Адрес - 23296 Комментарий: Количество символов

после' REM, Имя - cbr code Длина - 1 Адрес - 233оо Комментарий: Код символов после REM, Вызов программы:

RANDOMIZE USR адрес контроль ошибок:

Если номер строки равен о, больше 9999. или строка с тем , же самый номером ухе существует, программа возвращается в BASIC. Комментарии:

эта программа не проверяет, достаточно ли свободной памяти для добавления новой строки. Следовательно, это должно быть сделано перед стартом с помощыэ программы "Размер свободной памяти" из нашей статьи. Символы для ввода после вей должны иметь йоды больие. чем 31, т.к. управляющие символа "(0-31) могут сбить с толку подпрограмму LIST в пзу.

Подпрограмма пзу. которая -вызывается для вставки символов. довольно медленно выполняется. занимая много времени.

Созданное выражение REM может быть использовано для хранения машинного кода или данных.

```
ЛИСТИНГ НАШШННХ КОДОВ НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,(23296) 42 0 91
            LD A. H 124
            OR1. 181
            RET. Z 200
            LD DE, 10000 17 16 39
            AND A 167
            SBC HL,DE 237 82
KET KC 208
            ADD HL, DE 25
            PUSH HL 229
            CALL 6510 205 110 25
JK HZ. CREATE 32 B
            POP HL 225
            RET 201
CREATE U) BC, C23296) Z3T 75 . 2 91
            PUSH BC 197
PVSH BC 197
LOA, 13 62 13
            CALL 3376 г05 136 15
IBC HL 35
            POP BC 193
8XT_CH POSH BC 197
            LD A. B 1HO
OB C 1T7
Jit Z. IBSERT 40 11
            LD A,(23300) 58 4 91
            CALL 3976 205 136 15
            INC HL 35
            POP BC 193
            DEC BC 11
JB HXT_CH 24 240
INSERT РвР BC 193
            LD A,234 6E 234
            CALL 3976 E05 136 15
i HC HL 35
            POP BC 193
            INC BC 3
I BC BC 3
LDA.B 1ЙО
            PUSH BC 197
            CALL 3976 205 136 15
            POP BC 193
IH6< HL 35
            LD A. C 121 _
            CALL 3976 205 136 15
            INC HL 35
            POP BC 193
            LD A,C iJl
РЧ5Н BC ^ 197
            CALL 3976 * 205 136 15
```

```
            POP BC * 193
            INC HL 35
Li> A.F 120
JP 3976 195 136 15
```

Как она работает:

в пару регистров HL загружается заданная номер строки, это значение сравнивается с нулем, и, если он равен нулю, программа возвращается в бейсик. бсдя HL со-держит число бельке 9999 (максимальный возн&хныи номер строки) . также происходит возврат а

бейсик.

вызывается подпрограмма ПЗУ (CALL 6510), которая возвращает в HL адрес строки; номер которой бел в HL перед вызовом этой подпрограммы. Если Флаг нуля установлен, то строка уже существует, и в этом случае программа также возвращается в бейсик.

если Флаг нуля не установлен, делается переход к 'CREATE', в BC загружается количество символов для вставки после 'вен' и это число запоминается на стеке. в аккумулятор загружается число 13 код символа кнткк. Затем вызывается подпрограмма ПЗУ из адреса 3976 для вставки,символа ENTER. Регистр BC восстанавливается из стека.

после повторного сохранения BC на , стеке, пара BC тестируется, чтобя определить,, есть ли еще

символы для вставки. Вели нет, то делается переход к ' IKSERT*. Если есть еще символы для вставки, в аккумулятор загружается заданный нами код и подпрограмма по адресу 3976 (из пзу) используется для его вставки. Счетчик BC уменьшается, и программа возвращается к проверке BC на о.

Как только процедура достигает * IHSERT.', код йен вставляется, используя эту же подпрограмму. Затеи в BC загруяается длина созданной новой строки, и указатели длины для этой строки созданы. Номер строки затеи извлекается из стека, и это значение вставляется перед возвратом в бейсик.

7. 5 Сжатие программы

Длина: 71

количество переменных: о

Контрольная сумма: 715&

Назначение:

Эта программа удаляет все не-нужше управлявшие символы и свободное, пространства из бвйсш-програмны. увеличивая, таким образом, количество доступной ваня-

THv

Вызов процедуры:

RANDOMIZE USR адрес Контроль онибок:

Если в памяти нет беиснк-гао-граииы, процедура возвращается в бейсик. Комментарии:

эта процедура предполагает, что все выражения 8ен уже удалены из бейсик-программы. Если же это не так, произойдет сбой. Время выполнения пропорционально длине беисис-программы.

```
листинг МАЛИННЫХ кодов
МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,(23635) 42 83 9E
MEXT_L            INC HL 35
            INC HL 35
CHECK       LD DE,(23627) 237 91 75 93
            AND A 167
            SBC HL, DE 237 62
BET HC 208
            ADD HL, DE 25
LEHGTH            PUSH HL 229
            LD C, (HL) 78
            INC HL 35
```

```
            LD B, (HL) 70
NEXT_B            INC HL 35
LOAD        LD A,(HL) 126
                                    CP 13 254 13
J8 HZ.HUHBEB 32 8
REST        OR          POP HL 225
            LD (HL).C И3
            INC HL 35
            LD (HL).B 112
            ADD HL, BC 9
            INC HL 35
            JR KEXT.L 24 E27
HUHBER                                  CP 14 354 14
JB HZ, QUOTE 32 7
            INC HL 35
IIC HL 35
            INC HL 35
            INC HL 35
            INC HL 35
лг NEXT-B 24 231
QUOTE                               CP 34 254 34
лг HZ.CONTR зг 12
F.QUOT            INC HL 35
            LD A, (HL) 126
                                    CP 34 254 34
J» Z,NEXT_B 40 221
                                    CP 13 E54 13
JK Z. REST        OR40 223
Лг F-QUOT 24 244
COMTR                               CP 33 254 33
Л! HC. NEXT_B 46 211
            PUSH BC 197
            CALL 4120 205 24 16
            POP BC 193
            DEC BC И
            JR LOAD 24 204
```

как она работает:

в пару регистров HL загружается адрес бейсик-программы. HL затеи увеличивается дважды, указывая на два байта, хранящих длину следующей строки, в пару регистров DE загружается адрес области переменных. Бели HL не меньше, чем DE, подпрограмма возвращается в BASIC, т, к. достигнут конец программной области.

Адрес из HL сохраняется в стеке, в BC загружается длина строки, и HL увеличивается, указывая на следуший байт в строке. Байт в HL затем загружается в аккумулятор. Если это значение не равно 13, происходит переход к

'HUHBER'.

для достижения 'RESTOR' должен быть найден конец строки. Адрес указателей длины строки загружается из стека в HL и вставляется длина существующей строки, длина строки добавляется к HL, HL увеличивается и программа возврашз-ется к NEXT_L.

Если программа достигает 'HUKBER', проверяется, содержит ли аккумулятор символ нинвог (код 14). Если да, HL увеличивается на 5 - т. е. выполняется прыжок через следующее за HUHBEX число, и происходит переход к 'кехт BYTE'.

Если аккумулятор не содержит кода кавычки, программа переходит к 'сонтв*. если код кавычки найден, циклы программы будут выполняться до тех DOP, пока не будет достигнут конеп строки или найдена другая кавычка. в первом случае происходит переход к RESTOR. в последнем - к NEXT_в.

в процедуре CONTR происходит проверка символа. Если он имеет код не меньше, чем 33. подпрограмма возвращается к 'NEXT_в'.

Если свободное пространство или управляющий символ найдены.

вызывается подпрограмма ПЗУ по адресу 4120 для его удаления, длина строки, которая хранится в BC, уменьшается и происходит переход к 'LOAD'.

7. 6 Загрузка машинного кода в DATA-строку.

Длина: 179

Количество переменных: 2. Контрольная сунна: 19181 Назначение:

Эта программа создает выражение DATA в первой строке программы на БЕИСИЕе. а затеи заполняет его данными из памяти.

Переменные: Имя - data start Длина - 2 Адрес - 23296 комментарий: Адрес данных для

для копирования, имя - data length Длина - 2 Адрес - 83298 Комментарий: Количество байтов

для копирования. Вызов программы:

RANDOMIZE USR адрес контроль ошибок:

Если число байтов для копирования равно о. или первая строка уже есть, программа возвращается в бейсик. Программа не проверяет, достаточно ли памяти для новой строки. Программа требует 10 байтов на бант данных плюс пять байтов для номеров строк, указателей длины и т.д. кроме того, используемая подпрограмма ПЗУ также использует большую рабочую область - принимайте это в расчет. Если доступной памяти недостаточно, указатели строки будут установлены неправильно, и листинг БЕисика будет недостоверен.

Комментарий:

Время, занимаемое работой этой программы, пропорционально объему памяти для копирования.

```
ЛИСТИНГ НАШШНЫХ КОЛОВ МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD DE, (23296) 237 91 0 91
            LD BC, (23398) 237 75 2 91
            LD A. B 120
ОЙ С 177
            RET Z 200
            LD HL,(23635) 42 83 92
            LD A,(HL) 126
                        CP 0 254 0
            JR HZ.C08T 32 6
            INC HL 35
            LD A, (HL) 126
                        CP 1 254 1
            RET Z 200
            DEC HL 43
СОНТ        PUSH HL 229
            PUSH BC 197
            PUSH DE 213
            SUB A 155
            CALL 3976 205 136 15
            EX DE,HL 235
            LD A,1 62 1
            CALL 3976 205 136 15
            EX DE,HL 335
            CALL 3976 205 136 15
            EX DE,HL 235
            CALL 3976 205 136 15
            EX DE,HL 235
            LD A,228 62 228
            CALL 3976 205 136 15
            EX DE,HL 235
NEXT_B            POP DE 209
            LD A, <DE> 26
            PUSH DE 213
            LD C, 47 14 47
HUHDR       INC C 12
            LD B, 100 6 100
            SUB B 144
            JR HC, HUNDR 48 250
            ADD A, B 128
            LD B,A 71
            LD A. C 121
```

```
            PUSH BC 197
            CALL 3976 205 136 15
            EX DE,HL 235
            POP BC 193
            LD A.B 120
            LD C, 47 14 47
TEHS        INC C 12
            LD B. 10 610
            SUB B 144
            JR HC.TEHS 4ft 250
            ADD A, B 128
            LD B, A 71
            LD A,C 121
            PUSH BC 197
            CALL 3976 205 136 15
            POP BC 193
            EX DE,HL 335
            LD A, B 120
            ADD A,48 198 flfl
            CALL 3976 205 136 15
            EX DE,HL 235
            LD A, 14 62 14
            LD 8. 6 66
H_ZERO            PUSH BC 197
            CALL 3976 205 136 15
            POP BC 193
            EX DE,HL 235
            SUB A 151
            DJNZ M_ZERO 16 247
            POP DE 209
            PUSH HL 229
            DEC HL 43
            DEC HL 43
            DEC HL 43
            LD A, (DE) 26
            LD (HL),A 119
            POP HL 225
            INC DE 19
            POP BC 193
            DEC BC 11
            LD A, B 120
            ORC 177
            JR Z, ENTER 40 10
            PUSH BC 197
            PUSH DE 213
            LD A,44 62 44
            CALL 3976 205 136 15
            EX DE,HL 235
JB NEXT_B 24 17з
EHTES L.D A, 13 68 13
            CALL 39T& 205 136 15
            POP HL J25
            LD BC,O 10 0
            INC HL 35
            INC HL 35
            LD D.H 84
            LD E.L 93
            INC HL 35
POINT       INC HL 35
            INC BC 3
            LD A,(HL) 126
                                    CP 14 254- 14
            JR HZ.END? 32 12
            INC BC 3
            INC BC 3
```

```
            INC BC 3
            INC BC 3
            INC BC 3
            INC HL 35
            INC HL 35
I           INC HL 35
I           INC HL 35
'I IBC HL 35
I           JR POINT 24 237
IEND?                               CP ts 254 13
i|          JR KZ POINT 32 233 jl          LD A. C 121
II          LD (DE),A 18 li         INC DE 19 II            LD A. B 120
I           LD (DE).A id PET 201
```

как она работает: в бэру регистров DK загружает-I ся адрес байтов для копирования, I а в пару регистров BC загружается | число байтов для копирования. Ec-I ли BC содержит о, программа воз-Братается в вейсик.

II в пару регистров нь загружает-я ся адрес бейсик-программы, в ак-кунуяятор загружается байт из адреса, установленного в HL, это старший байт номера строки. Если он не равен о, первая строка не существует и программа переходит к 'COST'. если старился байт содержит о, в аккумулятор загружается младший байт. Если это значение установлено в 1, первая строка уже существует и программа возвращается в бейсик.

Адрес старшего байта номера строки сохраняется на стеке. Сохраняется число байтов для копирования . затем адрес данных. в аккумулятор загружается 0 (стзр-пшй байт нового номера строки). Вызванная подпрограмма пзу, нахо-дяшаяся по адресу 39'ft>. вставляет символ, имеющийся в аккумуляторе, по адресу, установленному в HL,

в HL восстанавливается значение, хранившееся там перед этой операцией. в аккумулятор загружается 1 и это значение вставляется 3 раза. Первая единица - это младший байт номера строки, следующие две - указатель длины

строки. 8 аккумулятор затем загружается код символа 'пата' и это значение ввводится в строку.

Адрес следующего байта давши восстанавливается из стека и загружается в DE, Сан этот байт загружается в аккумулятор, а содержимое DE вновь сохраняется на стеке. в с-регистр загружается значение, на 1 меньшее, чей код символа "о", с-регистр увеличивается, а в в-регистр загружается значение юо. в-регистр вычитается из аккумулятора и, если результат не отрицательный, подпрограмма возвращается к 'HUHDR'.

в-регистр прибавляется к аккумулятору (таким образом аккумулятор содержит положительное значение. Это значение затем загружается в в-регистр. в аккумулятор загружается содержимое с-регист-ра, а BC сохраняется на стеке. Вызовом подпрограммы ПЗУ (3976), вставляется символ, хранящийся в аккумуляторе, по адресу, содер-жашеиуся в HL, пара регистров BC восстанавливается из стека, а в аккумулятор загружается значение в-регистра. Выиеупонянут^й прием затеи повторяется для в=ю. аккумулятор затем увеличивается на 4в и полученный в результате символ вставляется в строку.

Вышеописанная подпрограмма делала вставку десятичного представления встречающегося байта данныг в выражение DATA. Теперь должно быть вставлено двоичное представление. Это значение маркируется с помошьс кода NUMBER 1CHR 14), который вводится первым, а за ним 5 нулей. Значение копируемого байта помешаетея в ячейку, чтобы заместить третий ноль. DE увеличивается, указывая на следующий байт данных. Число байтов для копирования снимается из стека в вс, и это значение уменьшается. Если результат равен о, делается переход к 'ENTER'. иначе содержимое пар регистров BC и DE вторично помешается в стек, в выражение DATA включается запятая, а программа возвращается к 'NEXT_в'.

в данной процедуре код ентех (конец строки) добавляется для маркировки конца выражения DATA. в HL загружается адрес начала строки, а BC устанавливается в о. HL увеличивается, указывая на младший байт указателя строки, и этот новый адрес копируется в DE, HL увеличивается, указывая на стзрвий байт указателя строки. HL и BC затем увеличиваются, а в ак-кумулятор загружается символ, наводившие я по адресу в HL,

Если аккунул-ятор евдвднт код 14, число найдено, то HL и BC увеличиваются на 5, перепрыгивая

через число и указывая на символ, следующий за ним. подпрограмма затем переходит к 'POINT'.

Если аккумулятор не содержит значений 14 и 13, происходит перевод к 'POIKT'.

Завершение Формирования строки выполняется помещением содержимого BC в ячейку указателя длины строки. На необходимый адрес указывает ое- по окончании работы процедура выполняет возврат в БЕЙСИК.

8. Инструментальные программы

8. 1 Определение размера свободной памяти.

Длина: 14

Количество переменных: О

Контрольная сумма: 1443

Назначение:

Дает количество свободного пространства ОЗУ в байтах. Вызов программы:

PRINT use адрес Контроль ошибок: нет Комментарий:

Эта программа должна вызываться перед использованием любых подпрограмм, которые могут увеличивать длину программы, чтобы бит уверенным в тон, что в ОЗУ достаточно свободного пространства.

листинг наминных кодов

```
НЕТКА АССЕМБЛЕР ЧИСЛА ¨ДЛЯ ВВОДА
            LD HL, О 33 О О
            ADD HL,SP 57
            LD DE, J23653) 237 91 101 92
            AND A 1&7
            SBC HL, DE 237 &2
            LD B. H 66
            LD C,L 77
            RET 201
```

Как она работает:

В пару регистров HL загружается о и это значение суммируется с адресом конца свободной области ОЗУ (адрес гранится в SP). Пара регистров DE загружается адресом начала свободной области ОЗУ и вычитается из HL, HL копируется в вс и программа возвращается в БЕЙСИК.

8. 2 Определение длины БЕЙСИК-программы.

Длина: 13

Количество переменных: о

Контрольная сумна: 1544 назначение:

дает длину бейсик-программы в байтах. Вызов программы:

P8INT USR адрес Контроль ошибок: Нет комментарий: Нет

ЛИСТИНГ МАШИННЫХ КОДОВ НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА

```
      LD HL, (23627) 4-2 75 92
      LD DE,(23635) 237 91 83 92
      AND A 167
      SBC HL, DE 237 82
U) B, H бв
      LD C, L 77
      RET 201
```

Как она работает:

в пару регистров HL загружается адрес области переменных, а в DE загружается адрес бейсик-прог-раины. DE вычитается из HL, чтобы получить длину программы. HL копируется в BC и программа возвращается в бейсик.

8.3 Определение адреса БЕЙСИК-строки.

Длина: 29

Количество переменных: 1

Контрольная сумма: 2351

Назначение:

Возвращает адрес первого символа после кода 'ЙЕН* в заданной строке.

Переменные: Имя - 1 те number длина - 2 Адрес - 23296

Комментарий: Номер строки, которая должна содержать

'ИНГ.

Вызов программы:

LET A - USR адрес контроль ошибок:

Если заданная строка не сушес-твует, или нет выражения REM. программа возвратит значение О.

Комментарии:

Эта программа может быть использована для нахождения адреса ячейки, в которую может быть дометен машинный код. после выполнения программы переменная а (может быть использована любая переменная) устанавливается по адресу, или в О, если имеет место окнбка.

Не вводить номер строки более 9999!

```
ЛИСТИНГ НАОИННЫХ КОДОВ МЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL,(23296) 42 0 91
            LD A,H 124
Ой L 1B1
            JR Z, ERR          OR40 5
            CALL 6510 805 110 25
            JR Z,COST 40 4
ERR         OR           LD BC,0 10 0
            RET 201
COHT        INC HL 35
            INC HL 35
            INC HL 35
            INC HL 35
            LD A,23J 62 234
                                  CP (HL) 190
JB HZ, ERR        OR32 243
            INC HL 35
            LD B,H 68
            LD C,L 77
            RET 201
```

Как она работает:

В пару регистров BL заносится определенный номер строки. Если этот номер равен о, делается переход к 'ERROR', иначе вызывается подпрограмма ПЗУ по адресу 6510. По возвращениии из этой процедуры HL содержит адрес строки. Если Флаг нуля установлен, делается переход к 'сонт*. если Флаг нуля не установлен, эта строка не суиествует, и подпрограмма переходит к 'ERROR', где в BC загружается о. и затеи происходит возврат в BASIC.

Если программа достигает метки 'COHT', HL увеличивается на 4. чтобы указать на первый оператор в данной строке. Если этот оператор не имеет кода 234. происходит переход к 'ERROR'. Если же это оператор 'REM'. HL увеличивается, указывая на следующий символ. Значение HL затем копируется в BC и программа возвращается в БЕЙСИК.

8.5 Копирование данных в памяти.

Длина: 33

количество перемевныk: з

Контрольная сумма: 4022

Назначение:

эта программа копирует область памяти с одного места в другое.

Переменные: Имя - start длина - 2

Адрес - 23296
Комментарий: адрес источника для копирования.
Имя - destination
Длина - 2
Адрес - 23298
Комментарии: адрес, в который происходит копирование.
Имя - Leneth
Длина - 2
Адрес - 23300
Комментарии: длина блока, подлежащего копирование. Вызов программы:
RANDOMIZE USR ЗАРес Контроль ошибок: Нет Комментарий:

Эта подпрограмма может быть использована для создания •мультипликации" с помощью следующего метода:

- создание первого экрана информации;
- копирование экрана выше RAMTOP;
- повторить для других экранов;
- копирование экранов в обратном направлении по одному в быстрой последовательности.

```
ЛИСТИНГ МАШИННЫХ КОДОВ НЕТКА АССЕМБЛЕР ЧИСЛА ДЛЯ ВВОДА
            LD HL, (23296) 42 0 91          LD DE, (23298) 237 91 2 91            LD
       BC,(23300) 237 75 4 91
            LD A, B 120
            ORC 177
            RET Z 200
            AND A 167
            SBC HL,DE 237 82
HET Z 200
            ADD HL, DE 25
            JR C.COPY 56 3
LDI» 237 176
BET 201
COPY        EX DE, HL 235
            ADD HL,BC 9
            EX DE, HL 235
            ADD HL,BC 9
            DEC HL 43
            DEC DE 27
LDDR 237 184
            RET 201
```

как она работает:

в пару регистров HL загружается адрес первого байта памяти для копирования, в DE загружается адрес, в который копируется память, а в BC загружается количество байтов для копирования. Если вс-о или HL^DE, то подпрограмма возвращается в бейсик. если HL бОЛЬше. чем DE, часть памяти копируется. используя инструкцию

'LDIR', и программа возвращается в бейсик.

Если же об больше, чем HL, то к обеим парам регистров прибавляется по вс-1 и память копируется, используя инструкцию 'LDDR', после чего программа возвраяается в бейсик.

Окончание следует.

ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ

КАНАЛЫ И ПОТОКИ

В 12-н номере -ZX-РЕВЮ" тл подняли вопрос о конвенции каналов и потоков. Среди наших читателей есть мнение о тон. что при всей нужности и важности этого вопроса, сама статья грешила определенной академичностью и, если читать ее от начала и до конца, то не все могли с ней разобраться. Большинство взяло то, что лежало на поверхности, а остальное отложили для разбора на долгое вреня,

в связи с этим мы с радостью принимаем предложение, высказанное наиин постоянный корреспондентом из г. Балашова Саратовской области Лашорииым в. г. о введении постоянной рубрики "Возвращаясь к напечатанному", и начинаем с того, что дадим рекомендации, которые предлагает наш уважаемый автор для тек, кто хотел бы не только в теории ознакомиться поближе с каналами и потоками, но и применить свои знания на практике.

Это достойное развитие начатой темы, и нам будет очень приятно. если полезное начинание будет подхвачено и новая рубрика вызовет появление новык публикаций.

для начала небольшое уточнение. Информация о существующих каналах хранится в ПЗУ и только после включения компьютера и инициализации системы эта ннФормация копируется в область, о которой указывалось в статье -непосредственно перед областью Бейсик-программы. а этот участок памяти, как известно, является уже частью ОЗУ. Такое дублирование области информации о каналах из ПЗУ в ОЗУ и одновременное существование ДВУХ одинаковых областей не является непродуманной расточительностью памяти, это глубоко продуманное решение, ориентированное на активных пользователей и разработчиков программного обеспечения, после инициализации система работает только с областью, наяодяшейся в ОЗУ. А как известно, из ОЗУ можно не только читать информацию, но и записывать туда свою информацию. Это значит, что внося продуманные изменения в существующую область информации о каналах иди расширяя ее для создания новых, пользова-тельстких, каналов, можно получить интересные эффекты, для примера выберем канал S. информация об этом канале находится в ячейках С 23739 ПО 23743 (СМ. ТЗбЛ. 1)

Таблица 1 Адрес Содержимое

?3739 244 9*256*244=2548 - адрес процедуры обслуживания

23740 9 канала при выводе

?3741 196 21*256*196-5572 - адрес процедуры обслужива-

23742 21 ння канала при вводе

?3743 83 код литеры s

во многих Фирменных программах для того, чтобы не искажать изображение на экране, полученное после дисплейного файла, перед загрузкой последующих блоков программы меняют адрес процедуры обслуживания канала s при выводе путем записи простых команд:

POKE 33739,B2

POKE 23740.O

ПРИ этом загрузка блока будет выполняться, но на экране не будет сопровождающих надписей:

Program...

Byte.. . и т. д.

Использование же команды CLOSE&2 для этих целей неприемлемо. После загрузки всех блоков программы адрес процедуры вывода вое станавлныветс я:

POKE 23739.244

POKE 23740.9

Этот нехитрый прием могут применять пользователи и для своих целей.

другое, не менее интересное применение концепции каналов и потоков для

практических целей -это изменение функции канала R. основное его назначение это ввод данных из буфера редактора, канал нельзя обслуживать, программируя на языке Бейсик, но его можно изменить так. чтобы редакция строк была невозможна, это будет интересно тем. кто работает над завитой своих программ от несанкционированного вмешательства. Достаточно записать:

POKE 23744. 124

POKE 23745,0

и редакция строки будет невозможна.

Аналогичные эффекты можно получить, манипулируя не с данными в области информации о каналах, а с данными в системной переменной STRWS (23566). так как каналы связаны с определенными потоками,

то переподключая потоки к другим каналам. получим новые эффекты. ну, например, чтобы запретить вывод на экран информации о программе при загрузке ее с магнитофона, достаточно записать в ячейку 23570 через команду роке вместо числа б число 16, переподключив поток-2. связанны» с каналом s к каналу р. обеспечивавшему вывод на принтер, переподключать с тандартные каналы к потокам можно и с помощыо hash-символа #. Попробуйте ввести программу (табл. 2):

и Вы убедитесь, что знакомые Вам Бейсик-инструкции работают несколько необычно.

кроне системных переменных

CHAHS{23631/2> и STRHS(23568), о

которых упоминалось в статье, су-шествует еще одна системная переменная (а о ней ничего не упоминалось), сохраняюоая информацию о каналах. Эта переменная CURCHL (23633/4) Current Channel, в которой записан адрес активизированного в данный момент канала. Именно значение этой переменной используется инструкцией RST 16 для вывода информации. Кроне того, системные переменные

FLAGS (23611), FLAGS 2(23658) и TV FLAG (23612)

также имеют некоторое отношение к каналам и потокам.

внимательный пользователь заметил, что для каналов к, s и р процедура вывода имеет один и тот же адрес - 2546. так вот, бит 1 переменной FLAGS указывает на то. должен ли символ выводится на принтер (бит равен 1) или на экран (бит равен о).

нулевой бит переменной TV FLAG информирует о тон, используется ли верхняя часть экрана (бит равен о) иди нижняя (бит равен 1). ну и бит 4 переменной FLAGS 2. находясь в различных состояниях.

таблица 2

ю PRIKT # о; "Текст в нижней части экрана": PAUSE о 20 LPRINT # 2; 'Текст а верхней части экрана': PAUSE о 30 LPRINT # з; -текст на принтере": PAUSE о

40 LIST # 1

(сброшен-установлен), определяет является ли канал к рабочий (бит установлен) или нерабочий в данный момент^(бит сброшен).

Прежде, чей перейти к процедурам, демонстрирующим возможность использования концепции каналов и потоков на'компьютерах с 4вК озу, еще одно замечание, поскольку, как отмечалось ваше, информация о каналах хранится и в ПЗУ и в озу, то можно, например, увеличить область Бейсик-про границ, исключив область информации о каналах из озу. Для этого надо переписать значение системных переменные: 1С роке 23635, реек 23631 20 роке 23636. реее 23638 30 роке 23631, 175 4о роке 23632. 21

здесь в строках ю и 20 меняется значение системной леренея-1ной, указывающей на начало Бей-1 с ик- про граммы. Теперь она будет Іначинаться не с адреса 23755, а с !адреса 23734. в строках же 30 и 40 в системную Беременную CHAHS записывается начальный адрес области информации о каналах, находящейся в пзу. Таким образом, участок памяти для хранения Бейсик-программы увеличивается на г! байт.

Весьма полезной, особенно для создателей обучающих программ, может быть следующая процедура, которая несколько меняет действие команды PRINT. При использовании этой процедуры по команде PRINT информация на экран будет выводиться с временной паузой между отдельными символами и с генерацией короткого звука после

вывода каждого символа, причем величина временной паузы между символами может быть эффективно использована при создании обучающих программ, например, по чтению или скорочтению. Эффект имитации работы пишущей машинки или телетайпа, кроме того, вносит разнообразие в работу программы (вспомните, например, игровую программу BlacK HawK).

А вот и сама процедура:

```
10 BEEP EQU 949
20 CURCHL EQU 23633
30 TIME EQU 65300
40 ORG &5301
50 PRINT DEFV 254в
60 START          PUSH AF
70          LD A. I
80          JR PO,KOPAUSE
90          LD A. (TIME)
100         LD B. A
110 PAUSE HALT
120         DJNZ PAUSE
130 HOPAUSE              POP AF
140                                 CP 33
1^0         JR C.HOBEEP
!60         PUSH AF
170         PUSH IX
180         LD DE. 50
190         LD HL, 100
200         CALL BEEP
210         POP IX
220         POP AF
230 MOBEEP        LD HL. (PRINT)
240         CALL ill
250         LD HL, (CURCHL)
260         LD BC, STAKT
270         LD E, (HL)
280         LD (HL),C
290         INC HL
300         LD D, (HU
310         LD (HL).B
320         LD A, B
330                                 CP D
340         JR HZ, CHHG
350         LD A, C
360                                 CP E
370         RET Z
380         LD (PRIBT).DE
390         RET
```

Запуск этой процедуры может показаться несколько необычным, так как для этого не используется функция вызова программы в машинных кодах USR. для того, чтобы эта процедура выполнялась, достаточно записать в область информации о каналах (в ячейки, отведенные для адреса процедуры вывода, обслуживающей канал S). адрес созданной процедуры. Стандартно тем записан адрес 2548, а мы с помощью команд росе записываем туда стартовый адрес созданной процедуры - 65303:

роке 23739, 23

роке 23740, 255

Теперь по команде RST 1 б (вызов процедуры вывода символа на экран! будет вызываться не процедура из ПЗУ а из адреса 65303/ временная пауза между выводимыми символами задается путем записи с помошью команды роке соответствующего числа {от 1 до ю) в ячейку 65300.

Пояснения к процедуре: в строка» 50... 80 выполняется проверка на разрешение прерывании. Если прерывания запрещены, то выполняется переход к строке с неткой HOPAUSE и обход инструкции HALT, приводящей к зависанию компьютера в этом случае.

- в строках 90... 120 Формируется требуемая временная пауза перед выводом на

экран очередного символа.

- в строках 130... 15о выполняется проверка кода выводимого символа. Если код менее 33, то это не печатаемый символ, а управляющий код и выполняется переход к строке с меткой новеер. минуя строки 160. . . 220. обеспечивающие выдачу звукового сигнала вместе с выводом символа на экран. Поскольку обработка управляющих кодов происходит с использованием другие процедур, то прямой вызов CALL 2548 может привести к ошибке или потере управляющего кода. поэтому выполняется вызов CALL ill. no

адресу in -записана команда JP (HL). такик образом мы реализуем несуществующую в ас с енблере процессора Z-80 команду CALL (HL) и выполняем перекод по адресу, указанному в регистровой паре HL (строка 13о).

внутри процедур обработки управляющих кодов имеются команды, которые меняют содержимое байтов области информации о каналах. поэтому в строках 250... 310 выполняется проверка записанного для канала S адреса процедуры вывода и восстановление адреса &5303, а затем восстановление! при необходимости. переменной PRINT (строки 320. ..390).

Предложенная Вашему вниманию процедура прекрасно работает только до момента, пока не будет выполнена команда CLS, так как эта команда восстанавливает з области информации о каналах стандартные адреса процедур вывода. Чтобы вернуть прежнее действие команде PRINT, необходимо опять же с помошью команд роке записать в ячейки 23739/40 адрес 65303. н так поступать после каждой команды CLS или нажатия клавиши ектек.

Естественно, это не совсем удобно. поэтому лучшим решением будет открытие нового пользовательского канала, адреса процедур ввода-вывода которого будут оставаться незыблемыми, пока включен компьютер. Для этого необходимо расширить область информации о каналах на 5 байтов, в которых зависать адреса соответствующих процедур ввода и вывода и имя нового канала. Выполняется это с помошью процедуры ПЗУ > находя-шейся по адресу 5717. при этом перед вызовом этой процедуры в регистровую пару HL необходимо записать начальный адрес резервируемой области, а в регистровую пару вс - количество резервируемых байтов. при использовании этой процедуры автоматически переписывается содержимое системных переменных, определяющих адреса отдельных блоков Бейсик-области. Ниже представлена программа, позволяющая создать новый канал: 10 FOR н-23296 то 23304: READ A:

роке н,а: NEXT н

20 DATA 1,5,0,33,202,92.195.85.22 30 RANDOMIZE USR 23296 40 RESTORE 60 ьо FOR H-23754 TO 23758: READ B:

POKE н. в: NEXT н 60 DATA 23,255,196.21,83 70 STOP

После выполнения этой программы останется только подключить канал к потоку 2 командой роке 23576.21. Теперь можно не бесоо коиться о необходимости восстанавливать адрес созданной процедуры в области данных о каналах.

Алексеев А., Г*.

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

Уважаемые читатели, если Вы читаете "ZX-ревю" не первый год, то должны быть знакомы с серией статей Стива Тернера под названием "Профессиональный подход". печатавшихся в прошлой году.

при всей уважении к английской школе программирования для "Сиек-труна" мы должны сказать, что и у нас немало хороших программистов. Только с кромность не позволяет честно признать, что для "Спек-труна" т нас их сейчас может быть уже и больюе. а не видно их работы только потону, что нет инфраструктуры - многочисленных фирм, способных приобретать их разработки и тиражировать нас совыми тиражами.

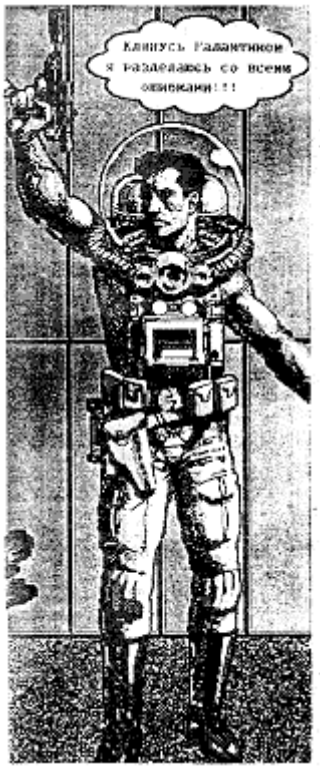
Нет пока у нас и средств массовой информации, сБОСобеих объединить миллионы людей, охваченных единой страстью. Мы прекрасно понимаем, что роль zx-PEBto а этом деле, конечно какая-то есть, но это не более, чек капля в море. Мы ведь не охватываем и сотой доли продента тех. кто в этом нуждается.

в нашем портфеле с каждым днем становится все больше и больие интересных статей от отечествен-еы8 программистов, который есть чем поделиться с многочисленными любителями, как с начинающими, так и с имеющими некоторый опыт.

Сегодня мы возвращаемся к рубрике "Профессиональный подход", но уже на другом уровне. Теперь мы открываем ее для наших отечественных специалистов. которым есть что сказать своим коллегам. Если и у Вас есть идеи и Вы можете их изложить доступно для понимания широкими массами, мы открыты и для вас. Работа оплачивается.

Обработка ошибок в БЕЙСИКе

Среди программистов, более или менее освоивших Бейсик, прошитый в ПЗУ "Ссектруна", бытует мнение, что невозможно на Бейсике создать хорошую "качественную" программу, которая к тону же выглядела бы достаточно "Фирменно". Хорошая программа - на 5 баллов - это, как правило, программа в машинных кодах. Существует также достаточно много расширений возможностей обычного Бейсика, это LASER -BASIC, BETA-BASIC. HEGA-BASIC и другие. Однако все ия надо догружать с ленты, при этом результн-руююая программа$получается достаточно громоздкой, эти расширения Бейсика хороши и оправданы



в тон случае, если Вы создаете, например, серьезную игровую программу, заведомо

претендующую на "фирменность". зачастую же, для повседневных задач не нужны богатые возможности расширений Бейсика. Прелесть Бейсика-пзу в тон, что он всегда сразу же готов к работе.

Что такое "повседневные задачи" ? ну, например, я применяю компьютер для обучения детей устному счету, обучения азбуке, одно дело, когда этону учит мама или папа, и совсем другое дело для ребенка вести диалог с компьюте-рон! эффект обучения здесь на порядок выше, компьютер может применяться в школе с первого до последнего класса. Ша эту тему можно говорить бесконечно.) а студентам очень пригодится, например. при расчетах курсовых

проектов. Это и есть "повседневные задачи", вы освоили Бейсик-пзу и в состояии» написать нехитрую программу для той задачи, которая сейчас нужна. Задача решена и программа, может быть, больше никогда не понадобится. а может быть Ваш труд и не пропадет даром, а заинтересует других. Но это произойдет вероятнее в том случае, если программа выглядит "Фирменно". Помните: встречают по одежке. я применяю несколько просты» приемов, которыми готов поделиться. Освоив ия. Вы сможете придать "Фирменный" вид любой сапой простои и примитивной Веи-сик-программе, Да Вы и сани нолу-и чите удовольствие от пользованияН такой программой. I!

II Итак, первый "Фирменный" прием. И

I

Предотвращение остановки БЕЙСИК-программ

Пало кому нравятся ьри с«к-программы, которые останавли ваются при первой нетретившеис нij ошибке или случайном нажатии кла-| виши "BREAK". Кроме неудобства вя расоте такие программы, выглядят! "иефирменно", в отличие от программ в машинных кодах, однако су чествует способ, при помони которого можно устранить остановку программы от клавиши "BREAK" или при встретившейся ошибке, причем в зависимости от ошибки и места в программе, где эта ошибка врой зошла. возможна различная. :iapa нее запланированная реакция прог раины.

Этот способ, по-моему, незаслуженно недостаточно рзспростpд мен среди ирогрднистов. хотя дает отличные результаты. Речь идет о программе в машинных кодах "он! ERROR GO то". Те. у кого есть!! Фирменная программа "SUPEKCOUK" I! или ее более поздние версии, воз-iI можно знают об упомянутом блоке)! кодов, но HP применяют его из за!' недостаточного удобства в работе. и Методика, изложенная ниже, позво I лит достаточно легко и эффективно! пользоваться этой программой, ав для тех. у кого нет программы!

"SUPERCODE", приводится блок ко"и дов 'ON ERROR GO TO". Для тех, кто интересуется програмированнен в машинных кодах, подробно описывается его работа, а если Вы не интересуетесь машинными кодами. то можете пропустить описание работы блока 'он кию» GO TO". IIPO-

читав Раздел о примере использования этого блока, вы сможете

"пристыковать" его к любой имеющейся г вас Бейсик-про грамме.

итак, сначала о блоке кодов -ON ERROR GO TO".

1. БЛОК КОДОВ *OH ERROR GO TO-.

Этот блок имеет длину 73 байта, он является редоцируеиым. то есть его можно загружать в любое место памяти, вызывается он обычным способом, то есть при помощи KAKDOHIZE USR ADDR. где ADDR -это начальный адрес загрузки едока кодов.

Рассмотрим работу блока, расположив его для примера в буфере принтера, начиная с адреса 23296 (#5BOO).

в листинге блока кодов справа число в скобках - это ссылки на номентарии по работе блока, описание которое дано ниже.

```
ТЕКСТ ПРОГРАММЫ •OH ERR       ORGO TO"
5B00 CD7C00             CALL #007C (1)
5BQ3 3B           DEC SP 12)
5804 3B           DEC SP
```

```
5805 El          POP HL (3)
SB06 010F00            LD BC.tt000F (4)
SB09 09          ADD HL. BC <5)
5B0A EB          EX DE.HL (6)
5B0B 2A3D5C            LD HL. (*5C3D) (7)
5B0E 73          LD (HL).E (8)
5B0F 33 1ЯС HL
5B10 72          LD (HL).D
5B11 C9          RET (9)
5B1E 3B          DEC SP (10)
5B13 3B          DEC SP
5BU совеог       CALL #огвJ mi
5B17 7B          LD A.E (12)
5B1* FEFF                                CP «FF
5B1A 20F8        JR HZ.S5B14 (13)
5B1C 3A3A5C            LD A. Ш5С3А) 114)
5B1F FEFF                                CP *FF
5B21 2821        JR Z,#5B44 (15)
5B23 FE07                                CP «07 (16)
5B25 2B1D        JR Z. «5B44
5B2T FE0e                                CP «0fi ti7)
5B39 3819        JR Z.K5B44
5B2B 3C          INC A (1C)
5BгС 32Й15С            LD (*5Св1).A (19)
5B2F FD3600FF          LD (IT+0). 4»FF(20)
5B33 210000            LD HL.tt0000 (21}
5B36 32425C            LD <«5С42). HLI22)
5B39 AF X        ORA (23)
5B9A Э2445С            LD <#5С44).A
5B3D FDCB01FE SET 7,(IY*1) (24)
5B41 C37D1B JP #1B7D (25)
5BW 33           INC SP (26)
5B45 33 IMC SP
5B46 C30313 JP #1303 (27)
```

БЛОК "ОН ERROR GO TO' состоит

иэ двух частей. Собственно программа по обработке ошибки начинается с адреса «5в12 (23314). а с адреса #5воо (23296) расположена процедура привязки блока кодов к

тем адресам, в которых он находится. Привязка осуществляется следующим образом.

сначала вызывается подпрограмма иэ ПЗУ (1). По указанному адресу «оотс расположена одна единственная команда: RET. при выполнении инструкции CALL на стек компьютера помешается адрес ячейки, в которое находится следующая за CALL команда, то есть в нашем случае «5воз (23299). а указатель стека - регистр SP -уменьшает свое значение на два.

Встречая инструкцию RET. выполнение программы продолжится с адреса, взятого со стека - это

#5воз. а указатель стека увеличит свое значение на два. Но число «5воз не стирается из памяти, чтобы возвратить его. производится уменьшение указателя стека на два (2) и число со стека записывается в регистр HL (3). (При этом указатель стека опять увеличивается па два. то есть возвращается к своему прежнему значению.) теперь в регистре HL

находится число #5воз 125299).

Далее (4) в регистр BC заносится величина смещения: ttO00F (015) байт и добавляется к содержимому регистра HL < 5 >. теперь в регистре HL будет #5воз+*000р-

=#5Bia <гЗг99*15=23314). это адрес начала программы по обработке ошибки.

Если расположить блок кодов

"он ERROR GO TO" с ЛЮбОГО другого

адреса, то в результате действий (1).. . (5) в регистре HL будет адрес начала программы по обработке ошибки, далее этот адрес пересылается из регистра HL в регистр DE (6) а в регистр HL загружается (т) двухбайтное число из ячеек

#5C3D, *5C3E (83613, 23&14). Это адрес системной переменной ERR SP. при ошибке

происходит переход на адрес. который содержится на стеке в ячейке, адрес которой находится в ERR SP. До работы программы "ON ERROR GO TO" там был адрес 4вб7 («1303) - это в ПЗУ подпрограмма вывода сообщения об ошибке. Этот адрес перехода подменяется на новый. то есть яа

#5613 (23314). записываемый ва стек побайтно из регистров е и D (в), после этого происходит возврат в Бейсик-программу (9).

на этом подготовительная часть закончена. Если ошибки при работе Бейсик-программы не происходит. то все работает обычным порядком, как и без программы 'ON ERROR GO то". а если происходит ошибка, то компьютер передает управление на адрес, записанный на стеке в варе ячеек, указанных в системной переменной ERR SP. то есть на

45в12. при этом происходит с лсд у-

соее.

Вначале указатель стека уменьшается на два. чтобы ве испортить стек при работе блока кодов "ON ERROR GO TO" (10). Затеи вызывается подпрограмма из ПЗУ "KEY-SCAS" (11). Результат действия этой подпрограммы отражается в регистре DE, Точнее, в регистре е - номер нажатой клавши. если не нажато ни одной клавиши, то в регистре е - #FF (255) (аналогич-но оператору IH в Бейсике).

Сравнивая содержимое регистра е с «FF, предварительно пересдав его (12) в аккумулятор а. определяется Факт нажатия на какую-либо клавишу. Если результат сравнения не ноль, то есть нажата какая-нибудь клавиша (это ножет быть, например, клавиша "BREAK"). то зацикливается подпрограмма "KET7-SCAH". приостанавливая дальнейшие действия. Как только клавиша будет отпущена, в регистре е появится число #FF и продолжится работа программы "ON ERROR GO

то".

Далее анализируется произошедшая ошибка, путей проверки содержимого ячейки системной переменной ERR SR, расположенной по адресу «5сза <23610). это код ошибки минус 1. Для этого содержимое ERR ев пересылается в аккумулятор (14). Если нет ошибки, а это может произойти в случае ло-гичного завершения программы и ее запланированной остановки в конце (о:ох), то есть код ошибки равен нулю, то содержимое ячейки 23610 будет 0-1 (что для одного байта эквивалентно 256-и то есть 255 (»FF). в этом случае управление будет передано (15) на адрес «5в44 (23364). Здесь (26) возвращается прежнее значение указателю стека SF и осуществляется переход на подпрограмму вывода сообщения об ошибке (27) (в налей случае это будет сообщение 0:ою, как это было бы при обычной работе Бейсик-программы.

Аналогичные действия будут и в случае появления ошибки с кодом в (в:еш) of file). (то есть рев 23610=7), такая ошибка может произойти при работе с внешней памятью, а также в случае появления ошибки с кодом 9 (9:STOP statement), (то есть реек гзбЮ:в), если в тексте программы стоит оператор STOP. Сравнивается содержимое аккумулятора с числом Тис числом вив случае совпадения, управление передается (16). <17) яа адрес #5844

(23364).

Дальнейшие действия являются подготовкой к запуску Бейсик-программы с заданной строки. они необходимы для корректной

дальаейшей работы интерпретатора Бейсика.

для этого прежде всего в ячейку, отведенную для системной переменной ERR до, заносится число »FF. имитируя отсутствие ошибки 20). Здесь следует пояснить, что ори работе интерпретатора Бейсика в "саектруме" в регистровой паре I? находится число #5сза (23610)

- адрес системной переменной EKS юг.

далее в два байта системной переменной HEW ppc (ее адрес -

-5С4в) записывается номер Бейсик-строки, на которую должен быть сделан переход. Для того, чтобы осуществить это, необходимо предварительно номер строки для перехода занести (21) в регистр HL и затем (22) содержимое HL переслать по адресу #5с48. здесь отметим для себя, что номер Бейсик-строки для перехода находится

в ячейках к5в34, #5в35 (23348,

23349). к этому моменту мы еще вернемся позже.

Затем обнуляется ячейка системной переменной HS ppc - это номер ораторa в строке, на которую будет сделан переход. Сначала обнуляется аккумулятор a (ез) и затем содержимое a пересылается в HS ppc - ячейку #5c44 123620).

теперь надо установить «лаг синтаксиса: 7-й разряд системной переменной FLAGS - управляющие Флаги Бейсика. имитируя положительный результат проверки бейсик

-строки на синтаксис (24). Адрес

FLAGS - «5сзв (23611) (или 1VM).

Теперь подготовка Бейсик-системы закончена и можно запускать интерпретатор Бейсика (25), Таким образом, Бейсик-программа будет запушена со строки, нонер которой задан двухбайтным числом по адресу «5в34 (23348).

Некоторые моменты в работе блока кодов могут показаться излишними, но благодаря им блок "ON ERROR GO то' устойчиво и надежно работает в разных режимах.

для того, чтобы получить блок кодов "ON ERROR GO TO", наберите следующую Бейсик-программу:

```
10 LET n=23296: LET S=0 20 FOR x^п то п*7J 30 READ y
40 роce X, г 50 LET S=S»y 60 NEXT X
70 IF 3<>729в тнен PRINT FLASH 1;"ERROR*: STOP 80 SAVE "on err" CODE 23296, 73 100 DATA
      205. 124,0,59.59.255. 1, 15. 0,9. 235. 42. 61. 92, 115.35, 114.201
110 DATA 59. 59. 205, 143. 2. 123, 254. 255, 32. 248, 58. 50. 92. 254. 255. 40. 33. 254.
      7. 40. 29. 254* 8, 40, 25
120 DATA 60.50.129.92.253,54,0, 255, 33, 0. 0. 34. 66, 93. 175, 50. 68. 92. 253.203,
      1,254
130 DATA 195.125.27.51,51,195,3 19
```

После старта, если Вы все набрали правильно, программа сформирует блок кодов и выдаст его для записи на магнитофон.

для того, чтобы показать работу программы 'ON ERROR GO то", нам понадобится демонстрационная Бейсик-программа, это будет программа под названеи "BEEPER", задача, решаемая ей. примитивна и не представляет практического интереса, но на этом примере будут продемонстрированы возможности ПРОГрамНЫ "ON ERROR GO TO" и Негоды ее практического использования.

2. программа 'BEEPER*. Текст программы:

```
1 GO то 100
2 BORDER 7: PAPER 7: IKK 0: C LS
3 LOAD "on err- CODE 23296
4 RUM
5-SAVE "BEEPER- LIHE 2:
6 SAVE "ОП err" CODE 23296. 73
7 STOP 20 I8PUT ;
22 PRINT #0;TAB 8; FLASH 1; " p ress any кет -24 PAUSE 0 26          RETURN
100 BORDER 1: PAPER 7: IK 0: C LS
110 PRINT AT 12.8; PAPER 2; 1ЯЖ 7; BRIGHT l;- BEEPER " 120 GO SUB 20
130 BORDER 7: CLS : PRINT AT 21 .0;
1000 BEH -beeper-loio INPUT "Enter tune (-60. ..69 >: -; t
1020 PHIBT -Tune;-;t 1030 PRINT «0iTAB 8: INVERSE lf
B E E P
1040 FOR П-1 TO 50 1050 BEEP .07. t 1060 NEXT П 1070 GO SUB 20 1080 GO TO 1000
```

после того, как вы наберете программу, запустите ее со второй строки: RUH 2 и загрузите блок кодов -on err* CODE. После окончания загрузки нажните -BREAK- я выгрузите готовую программу на деяту. выполнив: RUH 5.

Теперь можете командой RUH запустить программу сначала. После появления "заставки", нажните лю-6у» клавишу, в ответ на запрос о величине тона введите число от -60 до 69 и нажните любую клави-

шу, вы услышите прерывистый звук, продолжающийся около пяти секунд, затем программа зацикливается, запрашивая следующую величину тона.

Теперь несколько слов о ситуациях, при которых происходит остановка программы.

Остановка с сообщением "L BREAK into proeram" произойдет при нажатии на клавишу "BREAK" в режиме ожидания, когда появляется мигающая надпись "press any Key" или в

момент звукового сигнала. Остановка с сообщением *D BREAK -сонт repeats" произойдет, если нажать "BREAK" в момент ожидания загрузки блока кодов (строка 2). Если при загрузке блока кодов произойдет ошибка магнитофона, то программа остановится с сообшени-ен "н tape load me error". Программу можно остановить, если на запрос о величине тона вместо числа ввести букву, например а. ь,,.. в этом случае остановка произойдет с сообщением "2 Variable not found-. Если введенная величина тона будет ниже -60 или выше 69. то программа остановится при попытке выполнить оператор веер с сообщением "в integer out of ranee".

Чтобы запустить программу "ON ERROR GO то", изменим строку 1:

1 RANDOMIZE USR 23296: GO TO 100

Теперь при любой ошибке, произошедшей после команды 8UH. программа будет запускаться со строки, номер которой задан двухбайтным числом по адресу 23348. так как тан 0. то программа переза-пустится с начальной строки (в вашем случае с первой строки), теперь остановить программу невозможно, вы можете убедиться в том, что при любой попытке остановки программа просто перезапустите я заново.

Это не совсем тот эффект. который нам необходим, но на этом этапе становится ясно, что остановить программу после старта невозможно, можно только перезагрузить ее с ленты, а это неудобно при отладке программы. поэтому надо позаботиться о возможности остановки программы в процессе отладки. Это можно сделать, нэп рниер. введя такой "жучок":

25 IF INKEY»=-4" THE! STOP

Теперь, когда на экране появляется кнгашвая табличка "press any key-, то программу можно остановить, нажав клавишу "G",

Для того, чтобы инициировать программу "ON ERROR GO то*, надо, чтобы было выполнено: RANDOMIZE

PSR 23296. Эту команду можно подставить в начало строки, с которой происходит запуск программы (строка и. Далее, на разных этапах выполнения программы надо изменять содержимое ячеек 23346. 23349. чтобы управлять переходом но ошибке согласно задуманное логике, например, добавим строки:

115 роке 83348,24: роке 23349, 0 1000 роке 63348.242: роке 23349.3 1065 роке 23348.0: роке 23349,0

Строка 115 блокирует клавишу "BREAK", исключая остановку программы в режиме "заставки", возвращая ее все время на строку 24 -ожидание нажатия какой-вибгдь клавиши, строка юоо определяет возврат на строку 10ю при ошибке ввода данных 1реж 2334в+Е5о»реек гз349=ююх. Строка 1065 определяет возможность перезапуска программы сначала при нажатии клавиши "BREAK", когда появляется нигающая надпись "press any Key".

Теперь можно считать, что на добились требуемой логики работы программы. Предотвращается остановка программы ври ошибке ввода данных. Кроме того, можно прервать звуковой сигнал в момент ис-йполнения <когда появляется табличка " ваа$"). а когда исполнение закончится, можно вообше перезапустить программу, нажав "BREAK".

однако следует отметить еще один момент, определенное неудобство доставляет перевод номера строки в двухбайтную форму. Это приходится делать вручную. к тону же, если придется переделывать программу в процессе отладки, из меняя нумерагаоо строк, то опять приходитея пересчитывать данные для роке гззад, роке гзздо.

Задачу можно значительно уо-ростить. если всномнить о том, что двухбайтный конвертер в "Сиектруме" уже есть. Это одера TOP RAflDOMIZE. который задает начальное значение для функции случайной величины RSD. Подав команду RAKDOHIZE n, мы устававливаем системную переменную SEED, используемую для вычисления очередного значения функции RMD. SEED расположена по адресу 23670 и занимает два байта. в ячейке 23&70 находятся младший. а в ячейке 23671 - старший байты числа п. которое используется с оператором RANDOMIZE. подайте кокая ду RANDOMIZE loio. Теперь сделайте:

PRINT реее 23&7G PRIKT реек 2367!

получите 242 и 3.

Теперь усовершенствуем нашу программу, добавив строки:

-IO роке г334&. реек 23670: роке 23349, PEEK 23&71: RETUfiE

Изменим строки:

115 RANDOMIZE 24: GO SUB 10 1000 RANDOMIZE 101O: GO SUB 10 IO&5 RANDOMIZE 1: GO SUB 10

Теперь, с точки зрения оporpa-мирования. задача упростилась. Здесь только надо отметить, что нельзя подавать команду RANDOMIZE о, так как в этом случае переменная SEED принимает не значение о. а становится равной другой системной переменной FRAHES - счетчика кадров, обеспечивая практически случайное число. Вместо RANDOMIZE о можно подавать кантон IZE 1. так как программа все равно обычно начинается со строки 1. (Если же у Вас присутствует нулевая строка и Вы хотите стартовать именно с нее, то просто подставьте в требуемое место, как и Раньше. роке ?334", о: роке 23349, о.)

Сейчас можно удалить строку 25, которая нужн была для отладки программы. Теперь единственный путем- позволяющим остановить программу. является нажатие "BREAK" при загрузке с ленты, в тот момент, когда Бейсик-программа уже загружена и ожидается ввод блока кодов "on err" (строка 2).

при желании кожно ликвидировать и эту возможность. для этого бЛОК кодов "он SJROR GO TO" можно

расположить внутри Бейсик-нрог-раммы, в нулевой строке. Наберите:

1 кем

а после кен - 73 пробела или любых других символов. затем сделайте роке 23756,о. Первая строка стала нулевой. Теперь выполните:

LOAD "on err" CODS 23760.

затем измените строки:

г RANDOMIZE I: GO SUB 1о: GO TO 100

10 POKE 23812, PEEK 23670; POKE 23813, PEEK 23671: RANDOMIZE USR 23760: RETURN

Теперь, когда блок "ON ERROR GO TO" расположен в новом месте, (начиная с адреса 23760). ячейками, в которых задана строка для перехода ао ошибке, будут ?3812,

23813.

строки 1. 3. 6 - удалите. они больше не нужны, так как теперь наша программа состоит из одного блока вместо двух и " горячий" старт (RUH 2) совпадает с "холодным" стартом <RUH). в этом варианте изменен также способ инициирования блока кодов "ON ERROR GO то". Команда RAH&OHIZE USR 2376О помешена теперь не в первую {стартовую} строку, как раньше, а

в строку ю. поэтому включение блока в работу происходит теперь в любом случае, когда выполняется команда GO SUB 10. независимо от того, с какой строки вы запустите Вашу ирО;танну. такой вариант более универсален.

Нн рассмотрели работу блока кодов "OB ERROR GO TO* на примере маленькой программы. в больших программах может потребоваться более сложная логика перехода при ошибке. Для этого используйте ячейку системных переменных 23681, куда заносится код ошибки при работе блока "ON ERROR GO то". Анализируя PEEK 23641. Вы можете организовать переход на нужную строку, для примера измените строку tooo программы "BEEPER":

1000 RAHDOH1ZE 2000: GO SUB 10 Добавьте строки:

E000 IF PEEK 23661=И тнен INPUT ;: PRIST #0; PAPER 2; INK 7; BRI GHT U-WARHIHG: -60<Tun e<69 ": BEEP 1,0: PAUSE 100 201О GO TO 1010

Теперь в тон случае, если вводимая величина будет выходить за пределы -60... +69. появится предупредительная табличка со звуковым сигналом. Конечно, проверку на допустимые пределы логичнее организовать при помоши обычного Бейсика, просто этот пример показывает, как можно использовать код ошибки, хотя надо сказать, что в своих программах мне еяе ни разу не приходилось этим приемом пользоваться.

в заключение хочу сказать, что при отладке программ с блоком "ON ERROR GO TO" почаше делайте RUH 5 (сохранение программы на ленте). в результате ошибочных действий может случиться так. что Вы не сможете остановить программу. Тогда останется только

загрузить предыдущий вариант. Если предполагаются значительные изменения в программе, то временно в начало строки ю подставьте RETURN :

это отключит бДОК "он ERROR GO

то". Кроме того всегда предусматривайте "жучок" для возможности остановки программы типа строки 27 в программе "BEEPER".

в следующий раз мы поговорим о структуре бейснк-программ, о том. с чего практически начинать написание новой программы, то есть когда в уме иди на бумаге план уже достаточно "созрел" и Вы включили компьютер, чтобы взбивать текст программы, а также о том. как облегчить себе жизаь. предусмотрев элементарный сервис для себя.

FORUM

В 11-12 номере "ZX-PEBC" (стр. ?54) за прошлый год ни писали о существовании 'циклических" защит программ от копирования и упоминали заветы класса ALKATRAZ LOADER.

в ответ на эту занетку пришло нескольк писем. Победи, одержанные нашими читателями над. этим загрузчиком с помощью аппаратных средств мы рассматривать не будем, считая этот путь не относящимся к теме, а вот частный случай, с которым разобрался д. ко-

ронцвит из г. жуковского. моск.

обл. , мы рассмотрим.

Почему нас не интересуют аппаратные пути взлома и копирования программ? Дело в том. что с программистской точки зрения взлом ради взлома, копирование ради копирования не интересны, нас интересуют программистские приемы, новые методы, короче интересует все тоi на чем можно учиться и набирать опыт. и. если быть до конца честными, то надо признаться, что любые статьи, посвященные вскрытию программ и снятию защит в основном служат не тем, кто их снимает, а тем, кто их ставит и тем. кто учится программировать.

итак, по порядку, просматривая загрузчики, написанные Сергеем СКОРОбогатовым для дискоФнпиро-ванныя им программ winter Edition. Chase H. Q. . Asent X и др. . наш читатель столкнулся с листингом, который выглядел примерно так (сн. листинг 1, комментарии к листингу - наш, -инфорком"):

Декодирование вручную, с по-мошью HOHS3 показало, что то. что было абракадаброй, содержит блок, очень похожий на то. что Вы видите на листинге, изменился только "ключ" и конечно изменился адрес, загружаемый в регистровую пару HL, Дальнейшее декодирование открыло еще один аналогичный блок и т. д. Длина и структура всех трех просмотренных блоков были одинаковыми.

конечно, это лишь частный случай, но этот Факт позволил написать программу, которая все пос-ледуюшее декодирование выполнила автоматически. Конечно, в более общих случаях она не сработает, но сам принцип будет полезен на-

листинг 1

```
        LD нь.пп - загрузили адрес, с которого начинается декодируемый блок.
        LD вс, гш - длина этого блока (организовали счетчик). LOOP          LD A, <HL) -
  приняли байт для декодирования.
кок "ключ" - само декодирование.
        LD (HL),A - заслали декодированное значение на место АБРАКАДАБРЫ.
        INC HL - передни к очередному байту.
        DEC BC - уменьшили счетчик байтов на единицу.
        LD a, в - подготовка к проверке счетчика на ноль.
OR c - проверка счетчика на ноль.
        JR HZ, LOOP
абракадабра ___ ........ .......... .
```

листинг 2 I

```
AGAIN       LD IX (aa) - aa - адрес ячейки, в которой организовано хранение адреса начала
  "взламываемого" блока.
        LD L.(IX*!)- второй и третий байты исследуемого блока со-
        LD H. IIX+2) держат адрес декодируемого куска
        LD c,(1x*4>- пятый и шестой байты исследуемого блока со-
        LD D.(IX+5) держат длину декодируемого куска. LOOP LQ A.(HL) - приняли байт для
  декодирования.
```

```
XOR (IX+8) - само декодирование.
I          LD IHLKA - заслали декодированное значение на место. I I            INC HL -
      перешли к очередному байту. I
           DEC BC - уменьшили счетчик байтов на единицу
           LD a.в - подготовка к проверке счетчика на ноль,
OR с - проверка счетчика на ноль.
           JR HZ,LOOP - если не все биты декодированы, переход к деко дированию очередного
      байта.
           LD вс, 010H - длина декодирующего блоке - 16 байтов (о) он].
           ADD IX,вс - "перепрыгнув" через рассмотренный блок, вводим в IX адрес начала
      следующего декодирующего блока.
           LD <аа).1х - помешаем его адрес в ячейку с адресом аа.
           LD а,оеен - оеен - это (236 в десятиричной системе) код операции XOR.
ср (1х*7) - сравниваем содержимое восьмого байта нового декодирующего блока с кодом 236,
      ожидая, что тан будет стоять XOR. I
           JR ?.. AGAIN - если это так. то возврат к началу и декодиро I вание следующего
      блока. I
           RET - если нет, то возврат в вызывающую программу.
```

I Отсутствие XOR в данной позиции может говорить либо о тон, что все декодирование успешно за вершено, либо о тон, что защита сменила ирин- I цип кодирования и уже не использует XOR или использует не только XOR. Тогда надо остано- I виться и посмотреть, что же она использует и I по-возможности внести изменения в свою декоди руюшую процедуру.

чинаюшим, а по мере необходимости они смогут адаптировать метод к конкретным условиям.

Пример декодирующей процедуры приведен в листинге 2.

Адвентюрные игры.

Heavy on the Magic.

Свое исследование этой увлекательной игры, выпушенной фирмой GARGOYLE GAMES в 1986 году ЯРИслал наш читатель из Грозного Каракашез а. г. с его полезными советами, пос вишенными игре Sceptre of Bagdad Вы знакомы но прошлому выпуску ZX-PKBK.

Ему удалось пройти все игровое программное пространство до конца, он посетил все комнаты замка,

рался с программой полностью. есть еще нерешенные проблемы, может быть кто-то знает подходы и к ним? Наш корреспондент пользе вался некоторыми ходящими по рукам непроверенными описаниями и ряд вопросов связаны именно с ними.

Начнем с проблем, а потом перейден к достижениям.

в описанияЕ говорится о 21 типе монстров. Обойдя весь замок, насчитать такого количества не удалось, даже если к монстрам отнести демонов, огонь, воду и пер* сонажей, неперенешаюшихся в пространстве и не нападающих на главного героя.

в описаниях говорится о 280 предметах, которые можно исследовать. если считать все. включая двери, столы, камни, сталактиты, никак более ?50 не получается.

в описания к говорится о том, что надписи на стенах встречаются довольно часто. удалось увидеть только ОДНУ - в комнате слева от начальной. Более того, нет нигде обешанной комнаты со множеством дверей, где висит особый знак, прочитав который

можно погибнуть.

Вскрыв игру с помощыо COPY-COPY, удалось обнаружить следующие имена демонов: ASTABOT, ASHODEE, BELEZBAR и MAGOT, но ни в каких книгах заклятий о них не упоминается.

таким же путей взлома удалось найти слова: ADEPT/US MAJOR. ADEP-TUS MIHOR, HAGISTR TEHPLI, MAGUS. и невооруженным глазом видно, что эти слова обозначают магический ранг. Но нигде эти ранги не участвуют. Открыв все двери и обойдя все, что можно, герои повысил свой ранг от HEOPHYTE до PHILOSO-PHUS через ZELATOR и PRACTICUS. Причем последняя дверь открывается только при ранге PHII.OSOPHUS -не больше, не меньше.

Кто знает, где и как можно получить те ранги, которые были обнаружены при вскрытии программы? Отзовитесь! Может быть у кого-то есть фирменное описание игры?

Теперь несколько заметок для тех, кто еще не начал работать с этой увлекательной программой.

На самом первом экране Вы видите Аксила между двумя столами с книгами. Левая - отравлена, а вот правая - GRIMOIR ("Гринуар") содержит заклятья: BLAST. IHVOEE.

FREEZE.

с помошыо заклятья BLAST можно уничтожать монстров, населяющих замок. Впрочем, для борьбы с самыми крепкими из них. придется прибегать еще и к помощи специальных предметов.

к дверям. f около которых есть столы, надо подбирать ключи. Когда Аксил кладет на стол нужный ключ, дверь открывается.

к тем дверям, около которых есть знак в виде двух переплетенных колец, необходимо кроне ключа принести еще и мешочек с золотой

(BAG OF GOLD) и положить его нз

стол. Нешочки с золотом взаимозаменяемы, т. е. любой такой мешочек (кроне отравленного) иодойдет к любой такой двери.

Двери, возле которых есть охрана, открываются паролями.

Проход в следующие три двери повышает магический ранг:

ASK APEX - паРОЛЬ "D00R. SI-LEHCE" (комната ZELATOR3.

SEEK FIRE BIRD TO EHTEK B00K -пароль "D00R. LAZA" (комната PRACTICUS).

THE GREAT SIGH I IH FREE - na-POл ь "D00R. SOROHOROS• (комната PHILOSOPHOS).

Прочие двери:

CRY AMD ENTER D00R - пароль "D00R, WOLF" - ведет на первый этаж.

то ENTER IS MADNESS - пароль "D00R. LUHACY" - открывает дальнейшие глубины заика.

SAY NUMBER OF MAGIC WORDS -пароль -D00R. ELEVEN" - назначение неясно. пройдя эту дверь, Аксил исполняет какой-то победный танец и получает сообщение. вну-шаюшее уверенность в собственных силах: Well done! Axil the able you can made it for an exit.

Аналогична ей и дверь: EYE FOR AN EYE TO ENTER PARADISE - OH3 открывается паролем "D00R. LONG",

Интересна дверь PILE тонв ко-KEY. Для того, чтобы ее открыть. надо привлечь на помошь демона: INVOKE ASMODEE-, а затем дать ему команду "ASHODEE, D00R", - и он разрушит дверь.

теперь несколько слов о демонах. Каждому демону соответствует свой талисман. Поэтому, чтобы вызвать демона, надо выложить этот талисман в комнате. Если его вызвать без талисмана, он жестоко наказывает, отправляя героя в адские печи.

Для демона BELESBAR нужен талисман HAHTIS. функции этого де мона пока неясны, единственное, что удалось установить - он дублирует команду ехем1нё.

Демон HAGOT предпочитает SUH-FLOWER. Он знает, где в замке расположен какой-либо предмет.

Предмет демона ASTAROT - меч (SWORD). Кажется, это самый полезный из демонов. Он выполняет роль телепорта и перенесет Вас в названную Вами область замка. Правда, есть недостаток - за мечом приходится возвращаться пешком. поскольку второй раз его уже

не вызвать.

ASMODEE - самый злобный демон. его можно вызвать только в комнате, в которой лежит рубин (RUBY). Он реагирует только на команду D00R и разрушает дверь, если вы имеете ранг PHILOSOPHUS, в противном случае с ним лучше дела не иметь.

Вызывая демона. Вы должны пом-

нить, что он появляется точно над талисманом, поэтому лучше встать от него чуть подальше.

теперь ряд полезных советов.

1. с поношью CLASP можно пройти через огонь.

а, BOUGAT можно положить на место HUGGET.

3. с помошью HUGGET можно убить оборотня (WEREWOLF).

4. с помощью зеркала (MIRROR) можно пбедить Медузу (MEDUSA).

5. вампира побеждают честноком (GARLIC).

6. SLAT побеждает циклопа.

7. BALL можно положить на место PELLET.

8. с помошью PELLET можно победить SLUG.

9. Вместо яйца <EGG) положить

скорлупу (SHELL).

10. Яйдо используется следующим образом, в районе KIDUS на первом уровне замка, где живут циклопы, нужно положить яйпо в огонь и вызвать птнцу-феникс "NEST. PHO-EHIX". Феникс должен вам сообщить пароль LAZA.

и. гидру (HYDRA) можно пройти с помошыэ SNAKE.

12. Компоненты ULNA, HEAD, THIGH нужно сложить в котел и произнести заклятие "CAULDRON. ACHADV в котле возникает новый монстр: AU, холодный и мертвый. Он подсказывает пароль LAZA, который уже сообшал Феникс. Может быть это сбой в программе, связанный с неудачным снятием ее защиты? Может быть. AU должен сказать что-то другое, например пароль LONG, который был извлечен из программы просмотром COPY-COPY и подбором.

13. Так же. просмотром кода был получен и пароль SORONOROS, хотя можно предположить, что его можно было бы вывести и из надписи на стене:

14. Чтобы пройти через воду, служит заклятие "WATER, FALL".

15. Пройти через пропасть (CHASH) можно, имея FLASK.

16. в одной комнате двери не оказалось. Трое стражников сообщили, что южной двери не будет. пока не найдешь пароль с помощью ERLSTOHE, Спросив об этом у HAROMa, получаешь ясный ответ, что искать егс надо в районе PIT: seeK it in the Pit. PIT находится в замке на четвертой этаже, но

никакого ERLSTOME там нет. демоны

APEX и BELEZBAR вз вопрос Об этом

просят показать ни ERLSTOHE.

Однако, с помощью хитрости, пройти за несуществующую дверь все же удалось. Просмотр в COPY-COPY дал слово MCHGATE, которое до этого нигде не использовалось, сказав "ASTAROT, LICHGATE" в комнате с иечон, удалось попасть за эту дверь, так и не найдя ERLSTOHE.

Таким образом, был пройден весь замок: ч этажа, в х в комнат, но нет никакого выхода и нет естественного Финала, ножет быть, кто-то из читателей прошел игру до конца и сможет указать на ошибку в действиях, а может быть это сбой в программе, связанный с неудачным снятием защиты?

и еще: нигде не удалось использовать кость (RIB) и костя IBOHES - три штуки).

в заключение наш читатель просит начать мозговой штурм игры "DUH DARACH",

выпушенной той же фирмой GARGOYLE GAMES. "инфорком" присоединяется к этому пожеланию и обращает внимание на то. что есть еще две игры: TIR-KA HOG и HARSPORT, не менее интересные и тоже пока никем не освешенные.

Полезные советы.

В дрошлом году мы несколько раз писали об особенностях работы компьютеров "Дубна 48" (с. 115. 156). Мы просили тех читателей, которые найдут программные пути повышения совместимости этой популярной модели, поделиться своими достижениями со всеми любителями "Спектрума".

Суть проблемы, если Вы помните, состоит в том, что из-за нехватки быстродействующих микросхем памяти в этой модели был применен пропессор с пониженной частотой, для чего были изменены некоторые константы в процедурах пзу, управляющих загрузкой программ. в итоге программы, снабженные спепзагрузчиками, обходящими пзу. могут не загружаться.

Своим видением этой небольшой проблемы и своими подходами сегодня делится наш читатель из поселка Провидения магаданской обл. Михаил Владимирович лапырев.

Во-первых, есть копировщики. работающие на "Дубне". которые позволяют Вам откопировать программу.

1. Lady COPY.
2. COPY П13
3. COPY-COPY (Pirate 02)

эти копировщики, правда, не компрессирующие, но свою задачу выполняют хорошо.

и будет совсем прекрасно, если вы достанете турбо-копировщики:

1 TURBO-COPY.

2 TUBBO-CAC.

Эти копировщики - компрессирующие и на "Дубае" они работают, хотя надо знать один нюанс, после загрузки копировщика и выгода в стартовое окно, не спешите нажинать LOAD. Включите магнитофон с программой, которую Вы хотите переписать и дождитесь пилоттона. идущего перед блоком программы, и только потом нажинайте клавишу. не беспокойтесь, если между блоками копировщик будет принимать посторонний сигнал и выдавать со-обшение об ошибке, после начала следующего пилотсигнала снова нажмите LOAD и так далее, всю за-грузку в копировшик выполняйте в режиме "т".

а вот с выгрузкой придется немного помудрить. Если программа стандартная, то ее можно выгружать, как обычно, клавишей *a" (ABTOJ в режиме "т". Если же у программы свой загрузчик, то первый блок бейсик-загрузчика и за-грузчик машинного кода надо выгружать в режиме "т", а остальные блоки - в режиме "L". после такой переделки программа будет работать на компьютере "Дубна". Правда, возрастет примерно в г раза время загрузки и увеличится расход ленты. Надо также помнить, что переделанные таким образом программы уже не смогут работать на других машинах.

Таким приемом удалось адаптировать под "Дубну" многие из первоначально неработавших программ. Конечно, не все еще доведено до конца, например, не поддается переделке программа SP00KED, у которой нестандартный загрузчик выполняет как бы загрузку "с конца блока". Но это поле для новых экспериментов.

Советы и секреты.

CHROHOS - если в таблице результатов набрать YIHG IT BABY (причем обязательно заглавными буквами), то получите бесконечную энергию.

COBRA FORCE - если переназначить клавиши в игре на "SIHOH", Вы получите бесконечную жизнь.

GEHINI WINGS - пароли:

уровен г - EYEPLAHT

уровен 3 - UHATWALI.

уровен 4 - G00DHITE

уровен 5 - SCULLDUG

уровен б - WIGH00TH

уровен 7 - GKEEPISH
PIPEMAHIA - некоторые пароли:
уровен 5 - DISK
уровен 9 - SAIL
уровен 13- OKCE
уровен 17- BOPE
уровен 21- PEHS
уровен 25- SLIP
уровен 29- EACH
уровен зз- RISE
AFTERBURHER: POKE 37934,0: 37935.0: 37936, о
RET STORH - POKE 37337, 201
секретами поделился Фильков Андрей Павлович из Москвы.

Письмо читателя.

Здравствуйте, 'инфорком- . ! я прочитал Ваш трехтомник по! изучению и работе с машинным кодом. Пока я не ознакомился с ва-1 шим изданием, я без успеха паты кался на ьарьер машинного кода. Мне всего тринадцать лет. но даже мне было нетрудно вникнуть в осо-бенности процессора z-80.

я являюсь подписчиком "ZX ре-

•вю" на 91-92 годы. До того, как я не прочитал вашего издания, руб-1 рику "Машинные коды" я просто! пролистывал, уделяя ей ноль вни-1 мания. I

Большое спасибо от всех людей, I которые пользуются лэвм "ZX-Spec truffi". I

Коллекция игр у меня небольшая! и я очень люблю умные игры, такие! как SHERI.OCK. KNYGHT TYME и THк| HOBBIT. я очень хотел бы обрати-I ться к экспертам, чтобы они за-! молвили слово об игре DUH DARACH. I Сам же инею по ней очень мало! информации. !

P. S. я прошу Вас напечатать в I "ZX-ревю" немного информации. I

Существует издание BEEP-SHOW, I которое ежемесячно будет ко всем! приходить, если вы обратитесь по! адресу: ч1ь5ю, Астраханская об. . I г. Ахтубинск-б. жуковского 24-63,1 UHICAL-STUDIO- I

Этот журнал рассказывает о!
разных звуковых эффектах. Молча |
юая программа заговорит и запоет. I
король юра. г. Свердловск. I

Проблемы совместимости.

Нашим постоянным читателям знакомы работы полубарьева С. в. . направленные на повышение совместимости отечественных моделей Синклер-совместимых машин с их английским прародителем 'ZX-Spectrum" . "инфорком" традиционно рассматривает проблемы совместимости, как одни из самых основных и сегодня мы предлагаем Вашему

вниманию две статьи, посвященные версиям "Ленинград-1" и "Пентагон-48".

доработки портов ввода/ вывода в Sinclair zx-spectrum модели "Ленинград-1я (версия Зонова».

гх-Зрес1гшв-совместимый компьютер "Ленинград-1я (версия зо-нова) создавался. по всей види ности, как максимально простой и дешевый вариант машины (или как самый доступный по элементам), о чем свидетельствуют многие примененные в нем схемотехнические решения (совмещенное поле ОЗУ 4в Кбайт и многие другие). Вероятно, именно по этой причине, логика адресации портов ввода/вывода в данной модели упрошена до такой степени, что это вызывает во многих игровык программах малоприятные эффекты мерцания бордюрной рамки экрана в такт с музыкой или нзвуком выстрела, а иногда и пол-йку» несовместимость, выражающуюся Ив "зависании" программ сразу пос-Нле загрузки, или же в невознож-jt ности управления от KEHPSTOK-Ндхоистика. однако, путем незначительных доработок скены, эти недостатки можно полностью устранить.

чтобы Вам полностью понять смысл вводимых изменений, здесь приводится краткая

справка об устройстве аналогичных портов 1/0 Фирменного ZX-Spectrura чзк, совместимости с которым мы добиваемся:

в Фирменном компьютере для работы с периферийными устройствами используются следующие адреса:

254 (FE HEX) но вводу:

- клавиатура, ввод с магнитной ленты (далее НЛ!; по выводу: цвет бордюрной рамки дисплея, звуковые эффекты, запись на МЛ.

И 251 (FB HEX)

И обслуживание фирменного узкопе-Н чатного устройства zx-prini.er 1| или аналогичных ему Timex-2040 !| AlPhacom-32, Seicosha GP-50S. |I 31 (IF HEX) I интерфейс джойстика типа Кещр-I ston.

!! При этом имеются следующие Sособенности:

I 1) при работе с нортон 254 1 анализируете я только сигнал на линии ао адресной шины процессора. поэтому с тем же самым успехом можно обращаться и к любому другому четному порту (252. 250. ... о), поскольку при его выборке анализируется лишь разряд ао.

2) KEKFSTOH-джойстик активируется в том случае, когда в адресе считываемого порта а5-"о" и ао;"1". при этой на шину данных

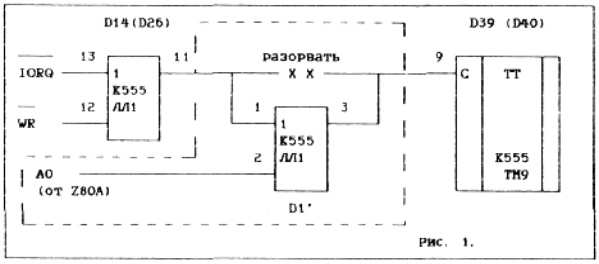


Рис. 1.

передается бант, младшие 5 бит которого зависят от положения

рукоятки и кнопки "FIRE" джойсти-

ка, а старшие 3 бита всегда равны "о". Таким образом, чтобы определить состояние джойстика типа KEHPSTOH, программа может обратиться к любому порту с нечетным адресом в диапазоне 1... 31.

3) если zx-prmter отсутствует в составе системы, то с порта 251 (#FB1 должен считываться байт 255 (ttFF). в противном случае компьютер будет неизбежно "зависать" при попытке выполнения операторов Бейсика LPRINT и LLIST. а также при загрузке некоторых игровых программ.

Теперь рассмотрим, как организована система ввода/вывода в версии "Ленинград-1":

1) по чтению порта 254 - совместимость с оригиналом полная, но запись в порт 254 выполняется при обращении к любому, а не только "четному" адресу порта (в этом причина мерцания бордюрной рамки). причина - при выборке иис к555тн9, на которой выполнен аорт вывода 254, адрес вообще не учитывается и не проверяется.

2) KEMPSTOH-джойстик имеет сразу Jдефекта: во-первых, считывается вообше по любому "нечетному" адрес у, во-вторых. 3 стар-яих бита установлены в "1" вместо

"о".

3) Вытекает из предыдущего пункта: поскольку джойстик считывается по любым, в том числе и не "своим" адресам, то из порта 251 может иногда читаться не совсем то, что надо бы. в авторском варианте схемы это проходит незамеченным, поскольку "1" в старших битах означают отсутствие принтера в системе, но стоит только "занудить" их. и периодические "зависания" машины вам надежно обеспечены.

Если Вы, прочитав вышеизложенное, узнали описание своих "бед", можете приступить к доработке. Остальным предлагаем убедиться в этом:

- выполните команду PRINT IH 31. На Фирменной машине должен быть получен результат о

(оо000000 в двоичной коде). Вы. скорее всего, получите 224 или 255.

- теперь выполните PBINT IH зз и PRINT in 251. Если результат первой команды совпадает с предыдущим - нарушена адресация джойстика. Если то же относится и ко ВТОРОЙ команде, да при этом еще и печатается не 255 - Вы уже наверное сталкивались,

иди скоро столкнетесь с проблемой загрузки игр. которые у ваших приятелей работали нормально.

- ВЫПОЛНИте OUT 255.о и OUT 255,

255. если ранка экрана стала при этом менять цвет - и здесь нарушена адресадия.

к счастью, разработчики платы этой машины предусмотрели в ее углу довольно большое макетное поле - "слепыш", на котором можно разместить все детали, необходимые для доработки.

Вам потребуются инс к555лл1 и к555ли1 (по 1 шт. каждого типа) и некоторое количество тонкого изолированного провода (лучше всего марки нгтф-0.07). дополнительные инс установите на свободное место на "слепыше" и подведите к ним питание (*5в) и "землю", теперь приступим к собственно доработкам:

1) Замените нагрузочные резисторы джойстика на мугг-о. 125. IKOM. их общую точку отсоедините от •+5в" и подключите к "земле". с "землей" следует также соединить выводы б, ю и 13 мультиплексора D37 ID42) к555кп11. Общий провод джойстика при этом соединяется с "+5в". это необходимо для использования джойстиков, изготовленных в стандарте "Atari", которые имеют нормально разомкнутые контакты ("Ленинград-1" спроектирован под самодельный джойстик с нормально замкнутыми контактами). Разумеется, что если ваш джойстик уже подключен через инверторы, то резисторы менять не обязательно, а свободные входы мультиплексора заземлить все равно придется.

примечаняе: автору известны 2 варианта чертежа принципиальной схемы, которые отличаются только нумерацией корпусов иис. поэтому

приводится двойная нумерация, вам следует определить, какая из них имеется у Вас. Для справки: упомянуло* мультиплексор установлен на плате рядом с разъемом кабеля клавиатуры, ближе к краю платы.

г} обеспечим стандартную адресацию порта вывода 254. выполненного на ймс D39 (&W) к555тн9. Для этого произведен следящие изменения в схеме компьютера (рис. 1):

Этим вы блокируете запись в регистр &555тн9 ори обращении по "нечетному" адресу, здесь и далее дополнительно введенные элементы имеют после номера значок " ' чтобы отличать их от элементов базовой схемы компьютера.

3) Обеспечим нормальную адресацию интерфейса Kempston-джойс-тика. Для этого служит следующая доработка (рис. 2):

при этой обеспечивается "Фирменная" выборка по следующему принципу:

- при ао-"о" - клавиатура;

- при а0="1" и а5="о" - джойстик!

- при ао-"1" и а5="1" - ничего не выбрано, при этом с шины данным считывается байт 255 (»FF).

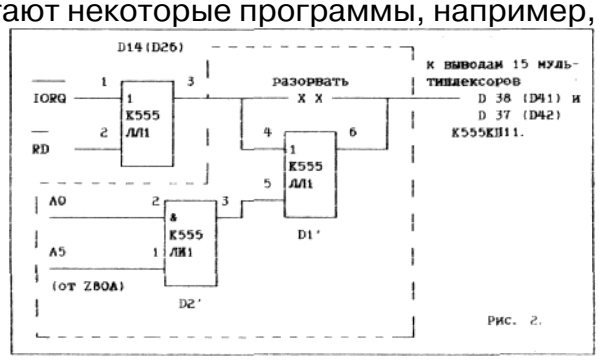
Вышеприведенные доработки не обеспечивают юо/. совместимости с Фирменным 'ZX-Spectrum (следует отметить, что абсолютно совместимых моделей самодельных спектру-нов на сегодня не существует. -причины этого достаточно сложны и их обсуждение не является задачей данной заметки), но позволяет ликвидировать подавляющее больший ство конфликтных ситуации между аппаратной частью и программным обесаечением компьютера.

Еше несколько примечаний. Во-первых, помните, что "Ленинград-1" - достаточно сложная машина, которую нетрудно вывести из строя неумелым вмешательством в схему и нелегко потом отремонтировать, поэтому не беритесь за доработку, если Вы не обладаете достаточный опытом работы с микропроцессорными системами, лучше обратитесь за помощью к более опытному человеку. Во-вторых, если Вас не волнуют проблемы с джойстиком, можете выполнить только пункт 2 (изменение выборки порта вывода 254). в-третьих, убедитесь перед началом доработки, что нонтая совпадает со схемой, и ранее до Вас никакого вмешательства в схему компьютера произведено не было.

Доработка "Pentagon-48K' для обеспечения совместимости с Sinclair "ZX-Spectrum"

На Spectrum в версии "Pentagon-sell", собранной в соответствии со схемой, не

работают некоторые программы, например,



"ELITE". копироввики COPY-86H. Outcopy. Одна из вероятных причин этого, возможно, заключается в следуювен.

Известно, что микропроцессор Z8OA имеет 3 режима обработки прерываний; IHO, IHI и IH2. с обработкой прерываний в первых двух режимах проблемы не возникает, поскольку и в тон, и в другом случае производится переход по Фиксированному адресу памяти, начиная с которого размешен обработчик прерывания. При обработке прерывания в режиме LH2 периферийное устройство, выдавшее сигнал 1нт. должно установить на шине данных младший байт адреса, по которому в памяти находится двухбайтный вектор адреса перехода на обработчик прерывании этого устройства. Этот байт считывается процессором. старший адресный байт, используемый для формирования полного адреса обработчика прерываний, берется процессором из регистра I. Таким .образом, в памяти может быть организована таблица векторов (точек входа в подпрограммы-драйверы внешних устройств), а сани драйверы располагаются произвольно, без привязки к жестко заданным аппаратной логиком адресам.

Хотя в компьютере ZX-Spectrum и не предусматривался режим 1н2, он вполне может быть использован при соблюдении определенных ограничений. Это основывается на тон. что в случае отсутствия на шине данных ао заданному адресу устройства, которое эти данные выставляет, с нее считывается байт

»FF (255), поскольку шинэ в ФИриеннон комиьютере "подтянута" резисторами в. 2 ком к "плюсу" источника питания. Единственным стандартным источником прерываний в spectrum является синхронизатор дисплея, выдающий 50 запросов на прерывание в секунду (в начале развертки каждого телевизионного кадра). Естественно, никаких данных на шину он при этом не вы-

ставляет. поэтому процессор примет #FF за младЕий байт адреса. по которому размешен вектор перехода на обработчик прерывания.

Таким образом, мы имеем возможность "перехватить" управление у стандартного драйвера прерывания, записанного в ПЗУ машины. для этого необходимо выполнить следуюаее: II

- разместить свой драйвер об л работки прерывания в произвольно выбранном Вани месте памяти;

- вектор перекода, по которому прроизводися передача управления драйверу, записать в озу. начиная с адреса axxFF IT, е. старший байт адреса произволен, а младший всегда равен 255);

зависать старший байт (#ях) в регистр I процессора;

выполнить команду ш2. Те перь при каждом прерывании будет запускаться не подпрограмма в пзу, а Ваш собственыи драйвер.

Такое решение может быть по лезным. например, если Вы пишете программу, в которой необходима постоянная индикация текущего времени на экране < типа "злектронных часов"), но параллельно с этим нужно выполнять и какие-то другие действия (скажем, редактировать текстовый документ), кото-в рые должны занимать основную^ часть процессорного времени. Воз-| можно применение такого способа и| при организации нестандартногои драйвера клавиатуры и для других •Фоновых" задач. Для любителей "покопаться" в коде фирменных программ не секрет, что немало игр могут использовать этот режим прерываний.

в "Pentaeon-48K" шина данных не имеет нагрузочных резисторов, в результате чего

при чтении байта с несуществующего устройства в! некоторые моменты времени ее сое к тояние может оказаться неопрелс и ленным. в режиме тг это может! привести к "захвату" ложного ад-й реса обработки прерывания, последствия чего очевидны, не исключено. что в игре "ELITE" иненно это и происходят,

Борьба с этим естественна -ввести в схему в резисторов, •подтягивающих" выводы DO... D8 процессора к шине "+5 вольт". Номинал этих резисторов для различных экземпляров машины может быть в пределах 3, 3. . . 10 кОн. Разумеется, не следует без достаточных оснований стараться установить резисторы "поменьше", поскольку это увеличивает коэффипи-ент нагрузки и без того перегруженного процессора! Попробуйте сперва т, 5.,. 10 кОн. и только если это не дает эффекта, уменьшайте сопротивление.

Для тек, кто имеет "Pentason-48K" без собранного контроллера "Beta Dish Interface", или с ним. но собранным по стандартной схеме (на плате установлены 4 ОЗУ к573рф4. рфь, J764 или е ПЗУ 27136 и шинный формирователь кр580ва86). пере дедки на этом з аканчиваютс я.

Тем же, кто использует в качестве ПЗУ микросхемы 27256, заменив микросхему кр580ва86 на перемычки, придется ее все же установить, иначе от перегрузки то и дело будет выходить из строя к561/ш1. установленная в контроллере. Но. впаивая кр5вова86. отсоедините ее вывод 9 (CS) от об-шего провода схемы и подайте на него сигнал IORO от процессора Z80A. Это устранит конфликт на шине данных между J7256 и ва86.

Вышеописанные доработки неоднократно проверены на практике и значительно улучшают совместимость "Pentagon-4flK".

На необходимость доработки "Ленинграда" по дешифрации внешних портов указывают и другие авторы, нн не будем здесь повторять все присланные рекомендации, но приведем несколько полезных советов от Иванова н. ю. из якутии, касающихся других вопросов.

1. RGB-выход на "Ленинграде". (Рис. 3).

Микросхема D11 (555лп5) желательно должна быть серии 555. серии же 155 и 1533 работают не вполне хорошо. D40 тоже желательно серии 555.

2. Если на ножку б процессора подать 5V через резистор 1 ХОн (обязательно по паспорту). то плата будет работать при пониженном напряжении от 4,5 до 5.5 вольт. Были даже случае и от 3. в до 4 в.

3. сигнал INT можно настраивать по музыке из программ Savaee или saboteur, эти программы очень чувствительны к длительности им-пульса, а недостаток этот присуш многим заводским машинам.

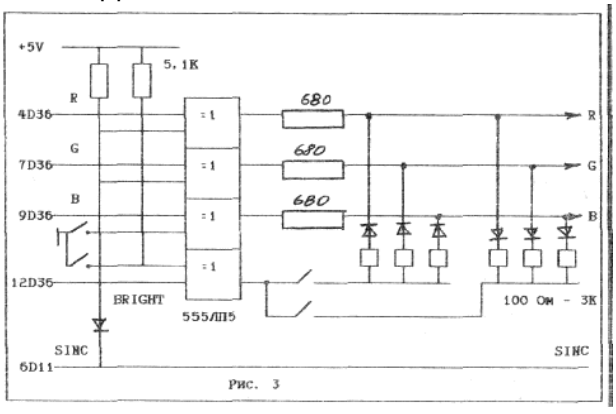


Рис. 3

Проблемы "ELITE".

Ны уже писали о том. что есть возможность перехвата "Таргонов" в гиперпространстве, нногие пилоты пользуются этим для быстрого повышения игрового рейтинга. к сожалению. ори этом могут возникать проблемы с нехваткой топлива на Вашем корабле.

Оригинальное решение проблемы нашли пилоты из Екатеринбурга

ДРемИН А. (DEADLY), кулик с. (DEADLY) и Киселев д. А. (ELITE) из Казани. Вот их рекомендации:

Перед боем надо заправиться топливом у звезды и оттуда сделать гиперпереход.

Следите, чтобы в этот момент надпись FUEL SC00PS он была на экране.

Эффект следующий, во время боев в гиперпространстве все время будет гореть эта надпись, как будто дозаправка топливом не прекращалась. Вы сможете переходить от одной атаки к другой, не боясь "застрять" в космосе. Кстати, это мероприятия одновременно предохранит Вас от утечек горючего при повреждениях в бою (FUEL LEAK!).

Серьезному исследованию подвергли наши читатели поведение недокументированных космических объектов, впервые о которых сообщил нам Кислов д. н. из г. Челябинска (ZX-ревю-Эг. с. J53). Напомним, что это крупные объекты не-правильной геометрической формы. выпускающие в случае нападения на них несколько малых боевых аппаратов.

Семья потомственных художников пзриловых из знаменитого города Палеха исследовала эти объекты вблизи и пришли к выводу, что они хоть и имеют не вполне регулярную форму, но зато этих форм три. Более того, они различаются даже в месторасположении боевого лазера (конечно это можно установить только побывав в ближнем бою с кораблем, почти вплотную).

Абсолютное большинство писем, которые пришли от пилотов, содержат мнение о том. что это астероид с "отЕельниками". Наиболее глубокое исследование подготовил Вилин Д. П. из г. Симбирска. Он подтверждает данные Кислова и значительно их дополняет. Не претендуя на истину в последней инстанции, он систематизировал и об-обшил данные и пришел к выводу, что "объект" обладает и свойствами астероида с поселившимися на нем "отшельниками" и свойствами "космической платформы", но Факты, говорящие за первую версию, значительно более весомы. д. п. Ей лин высказал еще предположение, что авторы программы включили в нее корабль, обладающий и теми и другими свойствами для того, чтобы в условиях ограниченной памяти дать максимальный простор для Фантазии играющих. поскольку это не первый "Фокус" фирмы, то в этой гипотезе что-то есть. Вспомните хотя бы о том, что три дополнительные миссии и связанное с ними оружие были недокунентирова- мы в официальной инструкции. Также были недокуметированы и ряд управляющих клавиш. совершенно ясно, что многое Фирма "спрятала" для исследователей, ход очень оригинальный, позволяющий поддерживать интерес к программе годами.

Наш корреспондент предлагает •хаккеран" включиться в исследование программы не только "снаружи", но и как бы "изнутри". Так. в качестве объекта первого удара им предлагается код програины где-то зачиная с адреса 5850с и до конца озу. Здесь находятся текстовые сообшения, в которых, возможно, кто-то и найдет что-либо интересное.

Сообшения закодированы очень распространенным способом, который применяете» во многий программах, особенно в адвентюрных.

Известно, что в английском языке многие слова состоят из типичных слогов, например таких, как OR, IR, еа и т. п. Это позволяет уменьшать размеры текстового блока от 1. 5 до 5 раз. в ELITE закодированы даже целые слова, повторяющиеся иного раз. к примеру, LASER. COHPUTE8 и ДРУГИе.

каждому такому слову или слогу присваивается код. например. присвоим слогу ей код 145, тогда в памяти к примеру вместо слова GLIDER можно будет разместить G, L, I. D. 145. Обычно кодируют 256 слов или слогов. число 256 принято потому, что в этом случае код будет занимать 1 байт.

Просмотр таких закодированных сообщений может быть трудным. Это не гладкий текст, а обрывки каких -то фраз, череда разных букв. Поэтому этот способ не только сжимает текст, но еще и делает его нечитаемым. д, п. Жилин исследовал текстовый блок "Элиты" и даже составил программу по его декодированию, но работу еще не закончил в связи с нехваткой времени. Если кто-то найдет там что-либо интересное, мы с удовольствием напечатаем, сам же корреспондент приводит значения некоторых вскрытых им кодов:

128 - AL 129 - LE 131 - GL 133 - ce 134 - BI 137 - ES t38 - AR 139 - на 140 - IH 141 - D1 142 - RE 145 - AT 144 - ER 146 - ES 14в - RA 149 - LA 150 - VE 151 - TI 152 - ED 153 - ой 154 - QU 155 - ан 156 - те 156 - RI 159 - ON 165 - SYSTEM 174 - UMIT 167 - LASER 203 - ST 206 - CARGO 215 - COPMPUTER 227 - LARGE Следует также добавить, что обычно в программах такие коды

начинаются с кзкого-то определенного числа, ведь надо же оставить еще место для таблицы символов, в стандартном знакогенераторе символы расположены, начиная с 32 по 127.

Очень часто в программах для увеличения скрытности текста этот отрезок смешается в какую-либо сторону, то есть коды символов специально делают несовпадающими со стандартными. Однако, к счастью, создатели "Элиты" этого не

сделали.

• Примечание 'ИНФОРКОМа-: «

• за примерами того, как кодаруется текст в игровых ПРОГрамная далеко ходить не надо. *

• этот прием чрезвычайно широко распространен. Но ведь КОДИРУют не только текст, но и ГРЗфвку. Ны как раз подготовили статью о том, как это делают на примере программы JET SET WILLY. в одном из ближайших выпусков напечатаем и Бы сно- яете очень элегантно хранить

я графику своих программ с малым расходом памяти. *

47-ая Галактика: тупик или дорога к новым мирам?

Ны неоднократно упоминали о появлении 47-ой галактики в версии Родионова. Большинство наших читателей полагают, что это дефект программы, вызванный неаккуратным снятием защиты, на это указывают и неадекватные суммы кредов на счетах и сумасшедшее вооружение, но если внешний дефект смог вызвать появление 47-ой галактики, то возникает вопрос: "а нельзя ли нормальным путем проникнуть в новые миры ?". По крайней мере в двух письмах начато исследование этого вопроса. Но начнем все по-порядку.

Несколько месяцев назад мы предложили читателям начать атаку на тот отгрузочный блок, который сохраняется на ленте при сохранении состояния программы, нногие уже в этом преуспели и мы об этом тоже писали, к чему же ведет это исследование дальше? Давайте посмотрим.

Сначала мы представим выводы, сделанные Балисом Линасом из Каунаса.

1... 11 - имя пилота в символах ASCII. Байт, равный нулю информирует об окончании имени. 12 •- правовой статус, о - clean 1.. .49 - offender; 50. -.255 - fugitive. Как видите, правовой статус -величина не дискретная. т. е. может постепенно ухудшаться или улучшатьс я.

13... 15 - боевой рейтинг. Значения этой трехбайтной переменной таковы:

O, O. O. . . 255. 8. O - HARMLESS

O, 9, O. . . 255. 16, O - K. HARMLESS

O. 17,0. . 255.32.0 - P00R

O. 33. O. . 255. 64. O - AVERAGE

O. 65,0. .255. 126. O - Да AVERAGE

0.129,0.255.255.2 - COHPETEHT

O, 0. 3. . . 255. 255, 10- DAHGEROUS O, O, И.. 255. 255, 25- DEADLV 0. 0. 26. . 255. 255, 255-ELITE 16. , . 17 - ?

16 - номер карты галактики (0 -первая, 1 - вторая и т. д. t.

19 - CASH

1. . . 255 - 100O - 255000 Сг.

20 - CASH

1. . . 255 - 256000 - 65280000СГ

21 - CASH

1. . .255 - O. 1 - 25. 5 Сг

22 - CASH

1. . . 255 - 25, б - 6528 СГ

Сумма денег образуется суммированием всех этих четырех байтов.

24. . - 40 - наличие груза в корабле (см. ZX-PEBC-92. с. 254).

41 - грузоподъемность корабля. ЕСЛИ здесь содержится о и в байтая 24. ..40 тоже нули, то в корабле находятся беженцы (REFUGEES).

42 - FROHT LASER:
1 - pulse laser:
2 - beam laser;
3 - military laser:
4 - mining laser.
43 - REAR LASER
44 - LEFT LASER
45 - RIGHT LASER
46 - байт первой миссии. Если не 0. то дают миссию.
47 - FUEL
48 - количество ракет.
49 - LARGE CARGO BAY (В номере ZX-PEND-92, с. 254 здесь и далее ошибка).
50 - E. C. M. SYSTEH
51 - ДОПОЛНИТеЛЬНЫИ PULSE LASER
52 - Дополнительный BEAH LASER
53 - FUEL SC00PS
54 - ESCAPE POD
55 - EHERGY BOHB
56 - EHERGY UNIT
57 - DOCKING COMPUTER
58 - GALACTIC HYPERDRIVE
59 - ДОПОЛНИТеЛЬНЫЙ MILITARY LAS.
60 - Дополнительный HIKING LASER
61. . .66 - данные по галактике, в
КОТОРОЙ Вы наколитесь,
67 - ?
68 - Координата Y курсора на большой карте галактики. 0 - вверху; 255-внизу.
69 - ?
70 - Координата X курсора на большой карте галактики, о - слева; 255 - справа.
71 - 74 - ???
75 - 91 - содержат информацию о наличии товаров на станции.
92 - ?
93 - наличие дополнительного
оружия на корабле:
64.. . 127 - CLOAKING DEVICE;
128. .191 - E. C. M. SYS. JAMMER;
392. . 255 - И ТО И ДРУГОе. 94...102 - ??

Теперь рассмотрим внимательно байты 61... 66. содержащие данные по галактике, в которой Вы нахо-даггесь-

	1	г	3	4	5	6	7	8
61	Т4	lie	41	,вг	164	73	146	37
62	90	180	105	2ю	165	75	150	45
63	72	144	33	бб	13г	9	1»	36
64	г	4	6	1&	зг	64	128	1
65	вз	166	77	154	53	106	212	169
66	183	111	ггг	189	1гз	246	237	219

Обратите внимание на то. что содержимое данных по каждой галактике можно получить логический удвоением данных по предыдущей галактике. Ны говорим именно о логической, а не об ари»нетачес-ком удвоении именно потону, что если ори удвоении о&разуется чис-1ло, большее чем 255, то 255 от 1него отбрасывается, а в таблицу (заносится остаток.

Таким образом, облетев все во-сень галактик и полетев дальше, Вы попадаете в ... первую галактику.

и вот что по этому поводу пишет наш читатель из Каунаса:

"Если изменить значение хотя бы одного байта, мы попадем в но-вую систему из восьми галактик и названия няанет не будут повто-Ьяться. возможно, что в игре есть jкакие то условия, позволяющие пе-

!релететь в другие систему, но я их пока не обнаружил, сколько веете галактик в игре, я даже не 1берус ь ответить. прошу всех о 1помощи! надо найти способ переде та в другие галактики' я уверен, что он 'есть, и большая вероятность, что тан скрывается планетаRAXXLA. "

ны не случайно приводим слова нашего читателя из Литвы дословно. у Вас есть прекрасная возможность сравнить его выводы с выводами, полученными в то же время, но на несколько тысяч километров вое точнее.

Семейный экипаж из села гайтер хабаровского края в составе трое-глазова Павла герасимовича и его сына Геры провел огронную многодневную работу. Облетев восемь галактик, не уклоняясь ни от одного боя и отгружаясь на каждой планете они тщательно исследовали юг-байтный блок и тоже раскрыли циклический характер изменения байтов с 61 по ьб при переходе от 1 -ои до 6-ои галактик. Но они пошли дальше и исследовали галактику N47, в которую на некоторых версиях "Элиты" можно попасть "нелегальным" путей, как оказалось, с ной тоже связана группа из вое ьни галактик, данные по которым представлены ниже:

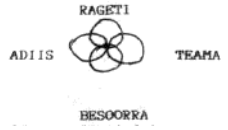
и вот. что пишут они:

"а где галактики между а-ой и 41-ой? Они есть: мы обнаружили за короткое время еще 36 галактик и прогулялись по ним. Яы также по дозреваем, что галактик может оказаться больше. чен 48... в глубинах этих неизвестных еще галактик модно. повидимону, повстречать и корабли поколении и пресловутую планету RAXXLA. ищите и обряяште '• ~

Вот как разыскивались эти новые галактики. В первоначальной варианте (система LAVE) исследо-

	41	42	43	44	45	46	47	48
61	129	3	6	12	24	48	96	192
62	161	67	134	13	26	52	104	206
63	245	235	215	175	95	190	125	250
64	72	144	33	66	132	9	18	36
65	80	160	65	130	5	10	20	4О
66	241	227	199	143	31	62	124	Л8

ватели обнуляли один за другим каждый байт из вести <&1. . .66) и каждый раз попадали в новую галактику. Пока еще неясно является ли каждая из вновь открытых галактик " опорной" для генерации целой серии из в ми галактик или эти галактики - часть одной системы. Невыяснены также вс е воп -росы, связанные с "устойчивостью" новых систем, дело в тон, что после перехода из первой во вторую и т. д. и вновь аосле возврата в первую могут наблюдаться сбои в работе программы. Возможно, есть какие-то законы, ограничивающие бесконечно возможное количество галактических систем, зато какие открытия тан возможны! Так. например, при б 1-ом байте, равном нулю, в саном венгре 5-ой галак тики есть четверная звезда:



Обнулив бг-ои байт и прогулявшись по полученной "восьмерке", наши корреспонденты с интересом увидели, что во второй галактике созвездия неожиданно похожи на земные: н.Медведица. Цефей, Дракон, Рыба. лира, третья галактика - необычно высоко сопиально раз вита, в шестой есть созвездие. похожее на Северную корону, а восьмая - кишит бандитами веек маетеИ.

вот к таким интересным открытиям может привести кропотливое исследование. Теперь дело совсем за малым - надо найти способ и условия возможности перемещения между галактическими системами. Но этот малый шаг можно пройти только объединив усилия всех пилотов и растянуться он может надолго.

авторы исследования полагают. что они затронули только "краешек" глобальной

проблемы довеле ния игры до логического конца.

-Новые" галактики - это еще не все интересное из того, что удалось установить семейному экипажу интерпретация байтов в основном та же, что и представленная выше, но зато есть некоторые любо пъггные комментарии.

1, Байты 24 - 4о - товары.

Здесь есть особешость. Максимальные цифры (255) можно вносить

только в 37-ой, зв-ой и 39-ый байты. Сумма же остальных байтов (товаров) не должна превышать 20 (35, если у Вас есть LARGE CARGO BAY). Все же, что выше этой нормы, будет при попытке продажи умножено на существующую в данной месте иену и списано с вашего банковского счета. а вот, что

будет, если Вы доведете свой счет таким способом до нуля и будете продавать дальше, - узнайте сани.

2. Байты 42... 44. засылая сюда число, большее чем 4. Вы получаете лазеры с самыми невероятными названиями. 99х их работают, как технологические и все имеют норо-шую скорострельность и огромную силу удара. Замечено, что борто-вые лазеры имеют примерно вдвое более высокую скорострельность, чем носовые и кормовые.

Интересно, что на этих "супер-лаэерах" можно хорошо подзаработать. при замене такого оружия на стандартное (из меню) на Ваш счет в банке поступает немалая суннз <30 тыс. и более) компенсации, но бывает и наоборот. Проверьте:

3. Байт 46 - миссия "Сверхновая".

Обычное значение - о. в зависимости от величины засланного сюда числа, вы получите миссию сразу или после нескольких перелетов с планеты на планету.

4. Интересно поэкспериментировать с байтами из группы 47-&о.

Засылая в чт-ой байт число 255, вы подучите на карте круг размером 25,5 световых лет и сможете перелететь на самую дальнюю планету этого круга за один обычный перелет, но если перелетите на ближайшую, излишки топлива у Вас отберут и останется обычный радиус в 7 световых лет.

заслав в 48-ой байт число юо или 200. вы получите соответственно юо или 200 ракет. при пус-ке ракет они могут появляться стаей на экране, заслоняя "пейзаж", можно стрелять очередями по 5-6 ракет.

Во все байты, кроне 50-го нож но засылать максимальное число -255.

5. С ГРУППОЙ &Я1ТОВ 94. . . 102.

УВЫ и ЭТОМУ экипажу справиться пока не удалось. Что ж. будем ждать новых ОТКРЫТИЙ.

I Кто может помочь с ремонтом I (японского дисковода TEAC *отзо- I │витесь. Вышла из СТРОЯ 40-ножеч-ная микросхема на плате 1553г097-05В.

7451оо, Туркмения, Балканская обл.. г. небнт-даг, кв-л 211,

Д.70. КВ 14, ВИННИКУ .

СОВЕТЫ ЭКСПЕРТОВ

Дорогие друзья!

Много читателей с интересом следят, как развивается "раскрутка" программы ELITE на страницах ZX-РЕВЮ. Вместе с тем, они справедливо отмечают, что давно бы пора найти этой "программе десятилетия" достойную замену и дружно предлагают на эту роль известную программу "ACADEMY" (развитие программы TAU CETI).

Сегодня мы предлагаем Вашему вниманию необычную экспертную проработку, выполненную сразу двумя авторами. Они проживают в разных городах и, возможно, даже не знают друг друга, но их объединяет любовь к этой программе. Мы же взяли на себя скромную задачу компиляции представленных ими материалов в единый блок.

Фамилии экспертов приведены в порядке поступления материала.

ACADEMY (TAU CETI II)

Автор: Pete Cooke.
Фирма: CRL Group PCL.
Год: 1986.

Эксперты:
Хоминич Р.В. , г.Киев
Жаров Р.Н. , г.Херсон

Академия Галактической Корпорации по повышению квалификации пилотов скиммеров (GASP) была основана в 2213 году после несчастного случая на 61 Cygnus, когда пилот новичок, выбрав неисправный аппарат, ошибочно произвел стыковку с реактором и половина планеты погибла под расплавленной лавой. Галактическая Корпорация приняла решение создать специальный тренировочный центр подготовки элитного корпуса пилотов, летающих на новейших военных скиммерах для применения в колониях и аванпостах Вселенной.

Выпускники академии направлялись в ряды специального корпуса скиммеров, где кандидатам требовалось успешно выполнить двадцать заданий, состоящих из пяти уровней, по четыре задания каждый.

Будьте внимательны! И, возможно, Вас будет ждать награда Галактической Корпорации.

Итак, прочитав сценарий игры, Вы нажимаете "FIRE" и переходите в главное меню.

Работа с меню.

Выбор осуществляется путем перемещения стрелки курсора и нажатием "FIRE". Выбранная Вами опция окрашивается в белый цвет.

Главное меню имеет следующие опции:
Accept Mission - выполнение миссии;
Select Mission - выбор миссии;
Select Skimmer - выбор скиммера;
Progress Report - рапорт о выполнении уровня;
Tape Menu - меню работы с лентой;
Enter a New Cadet - прием нового кадета;
View/Redefine Keys - просмотр/выбор клавиш;
Кроме этого в меню указаны: 00:00:00 - часы;
Sound - звук: "V"-вкл. , "x"-выкл;
Mission - текущая миссия;
Skimmer - выбранный скиммер;
Cadet - имя и фамилия кадета.

Просмотр/выбор клавиш.

Клавиши управления:
O - влево; P - вправо;
S - вверх/ускорение;
X - вниз/торможение;
N (SPACE) - огонь лазера/выбор;
M - огонь ракетой;
A - огонь противоракетным снарядом;
F - огонь осветительной ракетой;
B - сбрасывать бомбу;
V - круговой осмотр;
H - увеличение высоты;
G - уменьшение высоты;
J - прыжок;
L - приземление;
I - инфракрасное видение;
R - доклад о состоянии систем корабля.
Alter Keys - выбор клавиш: Kempston Joystick - джойстик;
Return To Main Menu - возврат в главное меню;
"BREAK" - восстановление стандартных клавиш.
Если у Вас подключен джойстик, то он будет выбран автоматически. Не рекомендуется менять клавиши управления (кроме первых пяти) во избежание путаницы.

Прием нового кадета.

Прошение о приеме в группу подготовки пилотов скиммеров.
Форма: VV1B/6702 (3 экземпляра).
Дата прошения 7/11/2047.
Имя (имя и фамилия): .....
Дата рождения (день/месяц/год): ../../....

Начать с I уровня (да/нет): ...

Для заполнения бланка введите свои имя и фамилию, а затем дату рождения. Год, видимо, надо рассчитывать, учитывая сегодняшнюю дату: 2047 год. Поэтому, если Вам 20 лет, то Вы родились в 2027 году и т. д. Вы можете выбрать любой год с 1901 ПО 2040. После введения даты в скобках появится день недели Вашего рождения.

Последний вопрос имеет смысл, если Вы находитесь на II-ом уровне или выше.

Меню работы с лентой.

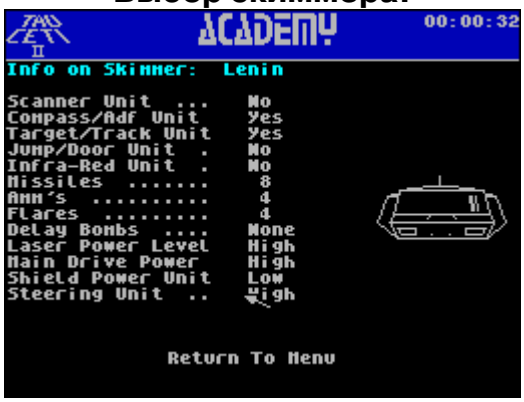
Load Game File - загрузить отложенную ситуацию.
Save Game File - записать текущую ситуацию.
Load Ship Designs - загрузить созданный корабль.
Save Ship Designs - сохранить созданный корабль.
Return To Main Menu - возврат в главное меню.

Рапорт о выполнении уровня.

В рапорте указаны имя кадета, уровень игры и степень выполнения каждой миссии.

Миссии считается выполненной, если Вы набрали не менее 90%. Уровень считается пройденный, если Вы набрали средний результат, близкий к 100%, выполнив все четыре задания.

Выбор скиммера.



Вы можете выбрать один из трех стандартных скиммеров:
GCS Lenin (Ленин),
GCS Lincoln (Линкольн),
GCS Wilson (Вильсон).

Кроме этого, Вы можете спроектировать скиммер по своему усмотрению (одновременно в памяти может быть три личных скиммера).

Тактико-технические характеристики стандартных скиммеров Вы можете просмотреть с помощью функций приведенных ниже:

Info on this Skimmer - информация о скиммере, отмеченном "V";
View Panel - просмотр панели скиммера;
Design New Skimmer - спроектировать новый скиммер;
Selection Complete - выбор завершен.

Проектирование скиммеров.

Для своего скиммера вы можете выбрать любое оборудование, соблюдая два условия: его стоимость не должна превышать 100 MCr и его вес не должен превышать 100 единиц.

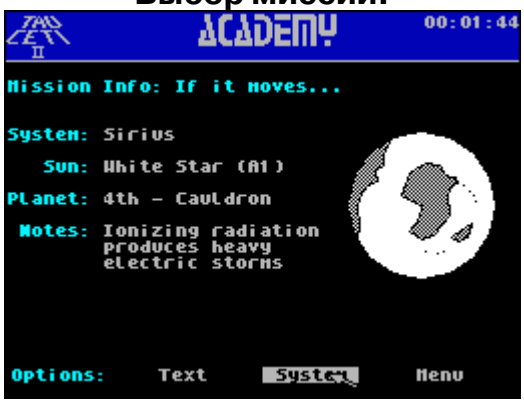
После выбора Design New Skimmer, перед Вами появится таблица:
Scanner Unit - сканнер;
Compass/Adf - компас/азимут;
Target/Track Unit - цель/курс;
Jump/Door Unit - гиперпрыжок/стыковка;
Infra-Red Unit - инфракрасное видение;

Missiles - ракеты;
Amm's - противоракетные снаряды;
Flares - осветительные ракеты;
Delay Bombs - бомбы замедленного действия;
Laser Power Level - уровень мощности лазера;
Main Drive Power - главный энергетический привод;
Shield Power Level - энергия защиты;
Steering Unit - блок управления.

Отметьте знаком "v" нужный ответ:
No (нет); Yes (да);
None (не требуется); 4 (штуки);
8 (штук); Low (низкий);
Med (средний); High (высокий).
После чего выберите Design Complete (проектирование завершено).
Если Вы решили отказаться от проекта, то выберите Abandon Design (отказ от проектирования).

Вы перешли в режим проектирования панели. Выберите структуру панели и ее цвет, затем расположите приборы (Place Instruments).
Появятся сообщения: Put Viewscreen (разместить обзорный экран) и Undo Last Step (отменить последнее действие).
Выберите место для экрана и поместите туда белый прямоугольник, нажав "FIRE". После чего выберите Menu (меню) и повторите операцию с окном сообщений (Message window). Желтый прямоугольник - поле занято, белый - поле свободно.
Далее размещается выбранное Вами оборудование. Последними размешаются: Height Gauge (шкала высоты), Shield Gauge (уровень защитной энергии), Fuel Gauge (шкала запаса топлива), Laser Temp (температура лазера) и Speed Gauge (шкала скорости).
На этом проектирование завершено. Осталось только присвоить имя новому скиммеру. При желании, Вы можете записать на ленту созданные корабли (см. меню работы с лентой).

Выбор миссии.



Для выбора миссии отметьте ее знаком "v". Используя Info in this Mission (информация о миссии), Вы сможете ознакомиться с целью (Text) и планетной системой (System), в которой Вам предстоит действовать.
Функция Load in Next Level (загрузить следующий уровень) будет заблокирована до полного выполнения всех миссий этого уровня.
Прежде, чем приступить к выполнению выбранной миссии, Вы должны внимательно изучить предоставленную информацию по звездной системе.

Уровень I.

Миссия I: If it moves... (Если оно движется...).
Цель: Уничтожение вторгшихся роботов.

Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков, нет системы поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Ленин.
Система: Sirius (Сириус).
Солнце: Белая звезда (A1).
Планета: 4-ая Cauldron (Котел).
Примечание: Ионизирующее излучение тяжелых элементов, электрические бури.

Миссия II: Red Dawn (Красный рассвет).
Цель: Уничтожение автоматических заводов во всех квадратах.
Счет очков: Процентное основание.
Примечание: Полная система гиперпрыжков и центр снабжения, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Вильсон.
Система: Betelguese (Бетельгейзе)
Солнце: Красный гигант (M2).
Планета: 6-ая Eventide (Вечер).
Примечание: Гигантское красное солнце, господствующее на небе, делает инфракрасную систему бесполезной.

Миссия III: Meltdown (Плавка).
Цель: Реактор в критическом состоянии - должен быть уничтожен.
Счет очков: 15 минут до расплавления.
Примечание: Нет системы гиперпрыжков, нет системы поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Ленин.
Система: Звезда Ван Маанена.
Солнце: Желтый карлик (dG5).
Планета: Escot.
Примечание: Одноликий мир, маленькая полярная колония.
Миссия IV: Softly Softly (Тихо-тихо).
Цель: Определить местонахождение и возвратится на базу.
Счет очков: По времени.
Примечание: Зона недавно заминирована оборонным сектором Галактической Корпорации. Причина минирования - административная ошибка.
Рекомендуемый скиммер: GCS Линкольн.
Система: Rigel (Ригель).
Солнце: Бело-голубая звезда (B8)
Планета: 12-ая Ice-world (Ледяной мир).
Примечание: Аванпост обороны, нет гражданских сооружений.

Уровень II.

Миссия I: Cipher (Шифр)
Цель: Собрать и смонтировать кодовые устройства реакторов.
Счет очков: Четыре кодовых блока.
Примечание: Нет системы гиперпрыжков, нет системы поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Линкольн.
Система: Vega (Вега).
Солнце: Белая звезда (A0).
Планета: 5-ая Homebase (Родная база)
Примечание: Маленькая, недавно сформированная колония.

Миссия II: At the OK Coral (В загоне все в порядке).
Цель: Уничтожение бродячих роботизированных систем.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков, система поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Ленин.
Система: Avior.
Солнце: Бело-голубая звезда (09)
Планета: 3-ая Corral (Кораль).
Примечание: Одноликий мир.

Миссия III: Where to Guv?
Цель: Пираты захватили гипертранспортную сеть - требуется их уничтожить.
Счет очков: Процентное основание.
Примечание: Полная система гиперпрыжков и система поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Линкольн.
Система: Sirius (Сириус).
Солнце: Белая звезда (A1).
Планета: Greengage (Зеленый заложник).
Примечание: Озоновый слой создает необычные световые эффекты.

Миссия IV: Hide and seek (Прятки).
Цель: Уничтожить комплекс солнечных дисков и вернуться на G.L.V. базу.
Счет очков: Процентное основание.
Примечание: Уничтожение дроидов может быть полезным.
Рекомендуемый скиммер: GCS Вильсон.
Система: Beta Hydri (Бета Гидры).
Солнце: Желтая (G1).
Планета: Engels (Энгельс).
Примечание: Одна из первых 4-х колоний.

Уровень III.

Миссия I: Laserium (Лазериум).
Цель: Уничтожение роботов-захватчиков.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков, небольшая поддержка, одна (1) G.L.V. база.
Система: Groombridge 34.
Солнце: Красная звезда (M2).
Планета: Единственная планета - Rayet (Лучистая).
Примечание: Малая новая колония.

Миссия II: Hades II (Гадес II).
Цель: Обнаружение и уничтожение вышедших из под контроля роботизированных систем.
Счет очков: Процентное основание.
Примечание: Небольшая система гиперпрыжков и центры поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Вильсон.
Система: Procyon (Проционн).
Солнце: Белая звезда (F5).
Планета: 15-ая Hades II.
Примечание: Очень удалена от солнца и постоянно покрыта льдом.

Миссия III: The Sands of Time (Песок времени).

Цель: Сеть реакторов должна быть уничтожена до эвакуации планеты.
Счет очков: Временное основание (1 час).
Примечание: Нет системы гиперпрыжков, нет системы поддержки, одна (1) G.L.V. база.
Рекомендуемый скиммер: GCS Ленин.
Система: Fomalhaut (Фомальгаут).
Солнце: Белая звезда (A3).
Планета: Foma-3.
Примечание: Изолированный аванпост.

Миссия IV: Mission Improbable (Миссия "Невероятная").
Цель: Собрать и смонтировать кодовые блоки реакторов и возвратиться на G.L.V. базу.
Счет очков: Процентное основание.
Примечание: Deaf-Фильтр применяется во многих реакторах.
Рекомендуемый скиммер: GCS Линкольн.
Система: Delta Pavonis (Дельта Павониса).
Солнце: Желтая (G1).
Планета: 3-я Dawnmist (Утренний туман).
Примечание: без примечаний.

Уровень IV.

Миссия I: Ceti Revisited (опять Тау Кита).
Цель: Сбор и монтаж кодовый блоков реакторов.
Счет очков: По мере сбора блоков.
Примечание: Система гиперпрыжков, небольшая поддержка, одна (1) G.L.V. база.
Система: Тау Кита.
Солнце: Желтая звезда (G8).
Планета: Третья планета.
Примечание: Одна из четырех первых колоний.

Миссия II: Out of the Frying Pan (Из огня …).
Цель: Обнаружение и уничтожение роботов вторжения.
Счет очков: Процентное основание.
Примечание: Небольшая система гиперпрыжков, полная поддержка, одна (1) G.L.V. база снабжения.
Система: Ригель.
Солнце: Бледно-голубая (O9)
Планета: 9-ая "Безымянная".
Примечание: нет примечаний.

Миссия III: Don't panic. (Без паники).
Цель: Уничтожение реакторов и солнечных дисков.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков, небольшая поддержка, одна (1) G.L.V. база.
Система: Эпсилон Инди.
Солнце: Оранжевая звезда (K5).
Планета: Adams.
Примечание: Радиация делает систему ночного зрения бесполезной.

Миссия IV: Needle in a Haystack (Иголка в стоге сена).
Цель: Определить собственное месторасположение и возвратиться на G.L.V. базу.
Счет очков: По времени.
Примечание: Предбазовая зона усеяна суперминами.

Система: Денебол.
Солнце: Белая (A3).
Планета: "Пыльный шар".
Примечание: без примечаний.

Уровень V.

Миссия I: Coal Mine (Угольные копи).
Цель: уничтожение враждебных роботов.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков и системы поддержки, одна (1) G.L.V. база.
Система: Epsilon Erandi.
Солнце: Красная звезда (K2).
Планета: Третья планета.
Примечание: Миссия на ночной стороне планеты.

Миссия II: PAZ!
Цель: Обнаружение и уничтожение роботов вторжения.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков и системы поддержки, одна (1) G.L.V. база. Роботы могут иметь суперракеты.
Система: Процион.
Солнце: Белая звезда (F5)
Планета: 7-ая "Нью-Марс".
Примечание: нет примечаний.

Миссия III: Protector. (Защитник).
Цель: Уничтожение следящей системы на Дельте.
Счет очков: Процентное основание.
Примечание: Нет системы гиперпрыжков и системы поддержки, одна (1) G.L.V. база снабжения.
Система: Альфа Центавра.
Солнце: Желтая звезда (G2).
Планета: Дельта.
Примечание: Ближайшая колония к Солнечной системе.

Миссия IV: The Shepherd (Пастух).
Цель: Найти и выстроить сторожевые башни рядом с G.L.V. - базой.
Счет очков: Пять сторожевых башен по мере сборки.
Примечание: Все пространство минировано.
Система: Альтаир.
Солнце: Белая (A7).
Планета: "Плато".
Примечание: без примечаний.

Полезные советы по сборке скиммеров.

Стандартные скиммеры не всегда являются удобными, поэтому рекомендуется пользоваться личными кораблями. Для проектирования корабля принимается во внимание наличие системы гиперпрыжков, системы поддержки, класс звезды и, конечно, Ваша цель. Возможно, Вам придется сделать несколько разведывательных вылетов, прежде чем Вы найдете приемлемый вариант.

Ниже приводится описание отдельных систем корабля.

Scanner Unit (радар) - рекомендуется ставить на все корабли, значительно облегчает выполнение задачи. Позволяет обнаруживать объекты и проходы в минных полях. Голубой квадрат в центре - зона непосредственного контакта. Граница квадрата - предельно допустимая дистанция приближения к минным полям.

Compass/Adf (Компас/УНБ Указатель Направления на Базу) - обязательный прибор (хотя в некоторых ситуациях опытный пилот может обойтись и без него). В случае его повреждения вдали от базы, Ваша гибель почти гарантирована. Азимут всегда указывает на Вашу G.L.V. базу.

Targeting/Tracking/Drain: Первый блок - это электронный прицел (может быть заменен или дополнен механическим).

Второй - настройка на радиомаяк.

Третий - указывает на то, что Вы подверглись нападению. Красный цвет индикатора указывает на то, что прибор включен. Последний блок может стать зеленого цвета (близкая опасность).

Jump/Door Unit - блок необходим при наличии системы гиперпрыжков или необходимости стыковки с другими объектами.

Система JUMP активизируется только в непосредственной близости от Jump-площадок. Запускается командой JUMP, красный цвет индикатора Jump указывает, что Вы вошли в зону действия платформы для гиперпрыжков.

Система DOOR нужна для открывания проходов в силовых полях.

Работает автоматически. Для открывания базы она не требуется. Красный цвет индикатора Door указывает, что стыковка запрещена (зеленый - разрешена).

Infra-Red Unit ,- прибор ночного видения. Дает контуры окружающих объектов. Необходим на планетах с резкой сменой дня и ночи (может быть заменен на осветительные ракеты). Включается клавишей Infra-red.

Missiles - самонаводящиеся ракеты (эффективное оружие). Пуск возможен только при наличии захвата цели. Пуск выполняется клавишей Fire miss.

Amn's - система противоракетной обороны. Энергетический залп возможен после получения сообщения от бортового компьютера о ракетной атаке. Пуск клавишей Fire amm.

Flares - осветительные ракеты. Пуск клавишей Fire Flar. Обеспечивают более хорошую видимость. По сравнению с инфракрасной системой, но имеют ограниченное время действия.

Delay Bombs - бомбы замедленного действия. Это чрезвычайно мощное оружие.

Уничтожают все объекты в радиусе действия, в том числе и скиммер. После сбрасывания требуется срочно покинуть заминированный квадрат.

Laser - основное оружие Вашего скиммера. Поражает практически все объекты, но при стрельбе быс-тро перегревается, поэтому его не следует использовать бесконтрольно.

Main Drive Power – главный энергопривод. Этот блок определяет максимальную скорость Вашего корабля, а также время ее достижения.

Shield Power Unit - отсек энергетической зашиты. Это генератор защитного поля. Определяет максимальную интенсивность поля и скорость ее восстановления до номинального уровня.

Steering Unit - блок управления. Определяет инерционность Вашего корабля в управлении.

Выполнение миссии.

Итак, цель определена, скиммер выбран и Вы приступили к выполнению миссии.

Перед вами появилась панель корабля, наверху указано положение скиммера, счет в % и бортовые часы. Компьютер дает Вам сообщение о том, что все системы готовы и просит ввести команду.

Наберите "HELP" для получения подсказки.

СВОДКА БОЕВЫХ КОМАНД

HELP - помощь;

LAUNCH - вылет скиммера;

PAUSE - пауза;

QUIT - выход в главное меню;

SIGHTS ON - прицелы включены;

SIGHTS OFF - прицелы выключены;

WAIT - ожидание (около 5 минут);

EQUIP - снаряжение скиммера;

STATUS - состояние систем корабля (аналог "R");

CODES - коды запирающей системы реактора;

DEAF - клавишный электронный звуковой фильтр;

LOOK – осмотр.

EQUIP - в этом режиме вы можете пополнить запасы оружия и произвести дозаправку (Refuel) и ремонт (Repair). Режим работает только в состоянии стыковки с базой или ремонтным центром.

STATUS - информация о состоянии бортового оборудования.

LOOK - информация о местонахождении корабля.

DEAF - включение Цифрового Электронного Аудио Фильтра, потренируйтесь с этим устройством. Оно сделано специально для зашиты от роботов, поскольку они страдают глухотой и дальтонизмом. Этот режим включается автоматически при попытке стыковки с объектом. Для нахождения пароля вам необходимо нажать Play и затем с помощью курсора повторить звуковую мелодию. В случае неудачи вы можете запросить другую мелодию с помощью Reset.

CODES - этот режим предназначен для монтажа кодовых блоков к реакторам. Используется в трех миссиях (2-1, 3-4, 4-1). Облетев реактор, Вы получаете определенное количество кодовых фишек. Обойму можно просмотреть в правом окне командами "вверх"/"вниз". После просмотра Вы с помощью команд Undo и Place переносите какой-либо блок в левое окно. Ваша задача совместить три кода для создания устойчивого рисунка. Изменяя положение и цвет (с помощью Flip и Colour) левого блока, Вы пытаетесь совместить его с каким-либо блоком из правого окна. (Этот этап практически является решением головоломки). Конечным результатом должно стать получение рисунка двух цифр, после чего первая часть кода будет зафиксирована в памяти. Следует повторять операцию до полного нахождения всех цифр.

Если все сделано правильно, код займет соответствующее положение, в противном

случае компьютер сообщит об ошибке и укажет на ее причину. Появившиеся цифры будут занесены в графу Locking System Code: _ _ _ _ _ _ _.

Миссия считается выполненной, если заполнены все прочерки.

(Кроме перечисленного компьютер реагирует на команду REACTOR, видимо уцелевшую от игры TAU CETI. При наборе команд, иногда допускаются грамматические ошибки).

Набрав команду LAUNCH, Вы оказываетесь на поверхности планеты и можете приступить к выполнению миссии. Во время боя компьютер информирует Вас о ракетной атаке, о применении роботами Amm's против Ваших ракет, об атаке камикадзе, о Ваших повреждениях, а также выдает некоторую другую информацию.

Итак, Ваша задача - получить звание пилота скиммера. Это произойдет, когда на экране Вашего компьютера появится надпись CADET QUALIFIED.

Если Вы более или менее научитесь управлять скиммером и перейдете к выполнению миссий, Вам могут пригодиться следующие полезные советы:

- КАТЕГОРИЧЕСКИ ЗАПРЕЩАЕТСЯ открывать огонь по платформам гиперпрыжков, центрам поддержки и реакторам! Автоматика ремонтных баз и реакторов, если Вы ее повредите, заблокирует проходы в силовом поле. JUMP-платформы легко повреждаются даже огнем лазера, но это сразу ставит под сомнение возможность выполнения миссии.

Практический опыт показывает, что без вреда (и без пользы) можно обстреливать только свои G.L.V. базы - они не повреждаются даже бомбой, но не пробуйте их таранить. База есть база. Стыковаться лучше на скорости не более, чем 1/4 от максимальной и ниже.

- не пытайтесь расстреливать минные поля - не хватит здоровья!

- не летите на полной скорости к неизвестному объекту (тем более к группе)!

- не забывайте своевременно возвращаться на базу для ремонта и заправки!

- при отмеченном большом количестве целей (радаром или визуально) желательно двигаться импульсами. Роботы обычно наступают шеренгами по 3-4 робота в каждой. Активизируйте шеренги по одной. Так Вы легко расправитесь с нападающими, но если Вы сразу активизируете 2-3 шеренги, Вам не удастся даже развернуться, чтобы отважно скрыться от погони.

- в некоторых миссиях при достаточном удалении от базы желательно фиксировать свое направление по компасу, а еще лучше - по Солнцу. Компас и радар - приборы очень хрупкие и в бою нередко ломаются, а без них, если Вы не знаете направления на базу, лучше начинать миссию заново.

- некоторые миссии Вы пройдете с первого раза, над некоторыми Вам придется поломать голову. Не отчаивайтесь и не ждите подсказок, из любой ситуации есть выход. Лежит он на поверхности, надо только его увидеть.

- помните, что при выходе в главное меню Ваши очки теряются!

Желаем Вам удачи!

SHERLOCK

Наши читатели помнят, как в шестом номере ZX-РЕВЮ за прошлый год мы давали полную фирменную инструкцию к программе "SHERLOCK". Программа заинтриговала многих наших читателей, но несмотря на дружные усилия до сих пор пока никому не удалось раскрыть это сложное и запутанное дело.

А дело действительно интересное. К счастью, совсем недавно, вытирая пыль в кабинете Холмса в доме на Бейкер-стрит 221 Б праправнучка миссис Хадсон случайно обнаружила пакет, датированный 1892 годом, на которой было написано "Вскрыть через сто лет в присутствии сэра К.Синклера и ответственного представителя "ИНФОРКОМа".

После вскрытия из конверта выпали пожелтевшие листочки. Давно выцветшие чернила и малоразборчивый почерк не помешали установить, что это заметки, которые велись по ходу расследования Лизерхэдской трагедии. К сожалению, судя по почерку, их писали не доктор Уотсон и не сам великий сыщик, так что никакой литературной обработки они не прошли. Сведения отрывочны, изложены небрежно. Кто этот неизвестный секретарь, посвященный в дела Холмса, еще предстоит разобраться биографам. Мы же спешим их опубликовать в надежде на то, что эти заметки помогут и Вам докопаться до сути самого захватывающего дела Холмса.

Понедельник, 8:00

Холмс и Уотсон в своей гостиной. По просьбе Холмса Уотсон раскрывает газету и находит в ней сообщение об убийстве молодой женщины в Лизерхэде (SAY TO WATSOH "READ CHROHICLE"). Холмс сразу же со всей присущей ему энергией начинает готовиться к расследованию дела.

Пройдя в гардеробную (OPEN PLAIN DOOR), он берет с вешалки два комплекта маскировочной одежды (DISGUISE). Первый комплект - для маскировки под китайца, второй - под старика. Чтобы взять костюм с собой, ему приходится сначала его надеть (WEAR), а затем снять (TAKE OFF). Вернувшись в комнату и поговорив с Уотсоном, Холмс направляется на место преступления и просит Уотсона следовать за ним (FOLLOW ME).

Ближайший поезд в Лизерхэд, как выясняется из расписания, отходит со станции Kings Cross, в 9 часов 15 минут. Чтобы успеть на него, Холмс, выйдя на улицу, должен взять извозчика (HAIL A CAB), сесть в экипаж (CLIMB INTO CAB) и объяснить ему, куда надо ехать (SAY TO CABBY "GO TO KINGS CROSS ROAD").

На третьей платформе Холмс встречается с инспектором Лестрейдом, который тоже следует на место преступления. Подождав до 9:15, сыщики садятся в поезд и в 10:30 прибывают в Лизерхэд.

Здесь Холмс предоставляет Лестрейду вести дело так, как ему кажется нужным. Он ходит за ним по городу (FOLLOW LESTRADE) и внимательно слушает все разговоры.

Расследование привело их к небольшому мосту, сделанному из песчаника. Здесь, на этом мосту и произошло преступление. Тело жертвы еще не убрали в ожидании полиции. Как Холмсу и Лестрейду удалось уже установить, это труп миссис Браун. Более внимательный осмотр места происшествия позволяет найти скомканную записку, подписанную инициалами T.F.

Миссис Браун была убита с близкого расстояния, пуля вошла в правый висок. Возможно, миссис Браун держала оружие в руках, поскольку на правой руке есть следы пороховых газов.

Холмс начинает подозревать, что здесь имело место не убийство, а самоубийство, у Лестрейда же определенно складывается иное мнение.

Проследовав за Лестрейдом к дому покойной. Холмс становится свидетелем допросов близких и домашней прислуги. Эти допросы помогли установить следующее:

- миссис Браун была вдовой недавно умершего мистера Брауна;

- мистер Браун был крупным ученым. Последнее время он работал над неким секретным военным проектом;

- после его смерти чертежи и прочая документация по проекту бесследно исчезли;

Кроме того, Холмс убеждается во время этих допросов, что мистеру Базилю Фиппсу доверять нельзя, несмотря на то, что он имеет твердое алиби на момент смерти миссис Браун (он весь вечер находился дома и играл на рояле этюды Шопена).

Кроме того, Холмс проводит беседу с майором Персивалем Фосом, проживающим на улице Sidmouth Street, который отказывается ему сообщить, где находился во время смерти миссис Браун.

Инспектор Лестрейд возвращается в Лондон. У него уже есть своя версия убийства и он подозревает в преступлении майора.

Обследуя дом, Холмс обнаруживает еще один труп. В библиотеке он находит тело миссис Джонс. Тщательно обследовав книжные шкафы, Холмс также обнаружил потайную комнату. В этой комнате хранилась женская одежда с пятнами крови. Изучение одежды и меток на ней показало, что принадлежит она Тришии Фендер (Tricia Fender).

При внимательном обследовании письменного стола (desk) в кабинете (study) Холмс обнаруживает ящик с двойным дном, здесь находятся банковские счета миссис Браун. Здесь же есть письмо от Тришии Фендер.

Судя по состоянию банковских счетов, миссис Браун в последнее время неоднократно снимала немалые суммы денег.

Теперь Холмс хочет поподробнее узнать о Тришии Фендер. Он возвращается в гостиную и продолжает допросы свидетелей (TELL ME ABOUT TRICIA FENDER).

Ему удается получить следующую информацию:

- Тришия Фендер была секретарем мистера Брауна;

- она проживает в Лондоне на улице Portman Street:

- она и миссис Джонс очень похожи друг на друга внешне.

Далее Холмс занимается проверкой свидетеля Базиля Фиппса. Он приходит к нему в квартиру на Cobden Lane и проходит наверх, в спальню. Здесь ему становится ясно, что ни на каком рояле свидетель не играл в ночь гибели миссис Браун. У него есть граммофон, на котором стоит пластинка с этюдами Шопена.

Из окна свисает простыня. Можно предположить, что кто-то использовал окно для выхода из дома.

Доверять Базилю Фиппсу нельзя, из свидетеля он превращается в подозреваемого.

Теперь Холмс имеет всю необходимую ему информацию и возвращается в Лондон.

Его ближайшая задача - допросить отставного майора, но сначала он забегает к Лестрейду в Скотланд-Ярд (на улице Parliament Street) и решительно заявляет ему, что майор невиновен (THE MAJOR IS INNOCENT).

Лестрейд требует доказательств, ведь майор отказался сообщить где он был во время убийства. Вместе они едут к майору домой на Sidmoutn Street.

Хозяина дома не оказалось и сыщикам пришлось ждать до 11 часов вечера, когда появился майор. Не задержавшись дома, он тут же снова вышел и, наняв кэб, приказал екать на Slater Street. Холмс и компания последовали за ним.

Слежка привела их к притону, в котором собираются курильщики опиума. Майор прошел внутрь. Для того, чтобы пройти за ним, Холмс должен переодеться в костюм китайца.

Теперь Холмсу понятно, почему майор скрывал, где он находился во время трагедии на мосту. Выйдя из курильни, Холмс говорит Лестрейду "THE MAJOR IS IN AN OPIUH DEN". Вскоре появляется и сам майор. Лестрейд отпускает его и говорит ХОЛМСУ "WELL DONE, HOLMES".

Так закончился первый день следствия.

Вторник.

Холмс возвращается в Лизерхэд и сразу же направляется к Базилю Фиппсу. Находит там комнату, в которой установлен сейф, вскрывает сейф и обнаруживает письма от Тришии Фейдер к миссис Браун. Из содержания писем становится ясным, что здесь имел место шантаж. По-видимому, Тришия похитила документы ученого и впоследствии шантажировала его вдову. На это же указывает и состояние банковских счетов миссис Браун.

Чтобы встретить Лестрейда, Холмс идет на станцию. Лестрейд прибывает около 9 часов утра. Здесь Холмс сообщает инспектору, что миссис Браун совершила самоубийство под влиянием шантажа: "MISSIS BROWN KILLED HERSELF".

Вместе они следуют на мост, ставший местом ее гибели. Тщательно обследовав ручей (CLOSELY EXAM THE DEAP STREAM), они находят пистолет, к которому привязан тяжелый камень. Очевидно, совершая самоубийство, миссис Браун хотела наказать шантажистку. Для этого она привязала пистолет к камню, чтобы после выстрела оружие исчезло навсегда. Для этого же она зажала в руке скомканную записку, наводящую на след T.F. То есть, она инсценировала убийство самой себя.

Мост сложен из мягкого камня, песчаника, и падая, пистолет оставил выщербину в нем. Именно это и побудило Холмса тщательно обследовать поток.

Теперь Холмс вновь идет на станцию, ему пора возвращаться в Лондон. Лестрейд соглашается, что миссис Браун совершила самоубийство и говорит: "WELL DONE, HOLMES".

Не позже, чем в 11:15 Холмс должен выехать в Лондон и прибыть на вокзал Кингс Кросс в 12:30. Наняв кэб, он едет на улицу Portman Road к дому, где живет Тришия.

Войдя в дом, он проходит в гостиную. У стены стоит сейф. Здесь его встречает хозяйка. Не обращая внимания на ее протесты, Холмс открывает сейф и находит папку с надписью "Военный Проект" и незаконченное письмо, обрывающееся словами: "Жди меня на мосту…".

Холмсу совершенно ясно, что именно Тришия Фендер шантажировала миссис Браун. Однако, в папке секретных документов не оказалось, очевидно они были похищены.

Холмс приказывает хозяйке дона следовать за ним, берет кэб и везет ее в Скотланд Ярд. Здесь они дожидаются прихода Лестрейда и Холмс приступает к допросу.

Подозреваемая сдается на вопросе об окровавленной одежде, найденной в доне миссис Браун (TELL ME ABOUT THE BLOODSTAINED CLOTHES) и начинает давать показания.

Сразу же открывается, что она вовсе и не Тришия Фендер, а настоящее ее имя - миссис Джонс. Она всегда была близкой подругой миссис Браун и, когда узнала, что та стала жертвой шантажа со стороны Тришии Фендер, убила шантажистку. Ее труп и был найден в библиотеке.

Теперь Холмс может сделать официальное заявление Лестрейду, как представителю власти: "MRS JONES KILLED TRICIA FENDER" ("миссис Джонс убила Тришию Фендер").

Лестрейду нужны неопровержимые улики и Холмс обращается к подозреваемой "TELL LESTRADE WHAT HAPPENED" ("расскажите Лестрейду о том, что произошло").

Если перед этим Холмс тщательно изучил все улики, не пропустил ничего важного, то она во всем сознается инспектору, придет дежурный полисмен и арестует преступницу. Если же это не произошло, значит что-то важное Холмс все-таки упустил.

После ареста миссис Джонс, Холмсу остается разрешить еще одну загадку - для кого же Тришия похитила секретные документы и где они сейчас находятся. После тщательного анализа всех собранных улик, Холмс приходит к выводу, что возможным преступником является Базиль Фиппс.

Из Скотланд Ярда Холмс отправляется на квартиру к Фиппсу, в Лондоне тот проживает в доме на улице Camden Street. Подождав под окнами до 10 часов вечера, можно дождаться, что кто-то изнутри откроет окно. Переодевшись в костюм старика. Холмс проникает в спальню через окно. Быстро осмотрев дом, Холмс остановил особое внимание на библиотеке. Здесь в мусорной корзине он находит клочки какой-то записки. На заднем дворе в мусорном ящике ему впоследствии тоже удается обнаружить порванную записку. Выбравшись из дома, переодевшись и изучив содержимое клочков, Холмс обнаруживает, что это обрывки одного зашифрованного послания. Недолго провозившись с простым шифром, он получает примерно следующее содержание:

"Чертежи у меня. Ваша цена меня устраивает. Прошу сообщить о времени покупки. Смерть миссис Браун вызвала вмешательство полиции. Базиль."

Для Холмса это неопровержимое свидетельство того, что похищенные документы находятся у Фиппса. С этим он и направляется в Скотланд Ярд, где до 7 часов утра ожидает прихода Лестрейда.

Среда.

Встретив Лестрейда, Холмс сообщает ему, у кого находятся чертежи: "BASIL HAS THE PLANS". Если к этому времени Холмс прочитал содержимое зашифрованной записки и обнаружил пустую папку из-под секретных документов, то Лестрейд предложит ему захватить преступников в момент передачи документов.

Холмс полагает, что сейчас Базиль Фиппс ожидает ответа от неизвестного сообщника и возвращается к дому на Camden Street.

В 9:5О появляется мальчик-посыльный и приносит Фиппсу записку. Холмс должен добыть ее любой ценой. Прокравшись к окну, он заглядывает внутрь комнаты. (LOOK THROUGH WINDOW). Если ничего интересного он не видит, наблюдение надо продолжать (команда повторяется). Наконец, где-то в 10:06 ему удается разглядеть горящий клочок бумаги. Теперь все решает скорость. Быстро влезть в окно. Схватить горящий лист и вылезти обратно. Маскировочный костюм теперь ему уже ни к чему.

Позже ему удается разобрать на обгоревшем листе зашифрованный текст, но на этот раз шифр уже другой. Кроме того, текст, за исключением имени адресата, написан задом наперед. Текст гласит:

"Базиль! Я покупаю чертежи. Буду в два тридцать на дороге Old Mill Road около Лезерхэда." Подписано инициалами H.W.

Быстро вернувшись в Скотланд Ярд, Холмс сообщает о том, что ему удалось узнать. Он отвечает на вопрос Лестрейда о том, где состоится передача секретных документов и они бросаются за преступниками.

Прибыв в 1:30 в Лизерхэд, Холмс выходит из поезда раньше Лестрейда и немедленно идет на главную улицу городка. Здесь уже ждет полицейский кэб. Задача Холмса успеть расположиться в кэбе до того, как его займут Лестрейд с помощником, иначе Холмсу просто не хватит места.

Лестрейд велит кэбмену ехать на Old Mill Road, они приезжают туда в 3:13, но это уже поздно. Базиль только что уехал. Лестрейд приказывет ехать обратно и в 4:47 они снова возвращаются на главную улицу. Здесь им удается заметить Баэиля и рядом с ним немецкого агента. Преследование приводит Холмса на платформу номер 2, где злоумышленники вскакивают в отходящий поезд. Холмс же должен найти другой путь в

Лондон. Сказав Лестрейду: "За мной!", (FOLLOW ME), он спешит в полицейский кэб. Уотсон также следует за ними. Холмс велит кэбмену ехать на King Cross Road.

Если все делать достаточно быстро, то Холмс имеет шанс увидеть, как Базиль и агент усаживаются в кэб. Удается и услышать, как Базиль приказывает кэбмену отправляться на Buckingham Palace Road.

Холмс мгновенно соображает, что соответствующая станция подземки - это Виктория (Victoria). Сыщики прыгают в поезд на платформе King Cross, который идет к Victoria, где и выходят.

Через некоторое время появляются преследуемые. Заметив Холмса, Базиль понимает, что он попался и пытается застрелить Холмса. Жизнь ему спасает доктор Уотсон (если он рядом). В последнюю секунду ему удается оттолкнуть Холмса в сторону.

Лестрейд производит арест.

Если Холмс сделает все это, то конечно он станет Величайшим сыщиком всего мира.

Так заканчивается одно из наиболее загадочный и интригующих расследований, которое вел когда-либо Шерлок Холмс. Почему оно не было до сих пор обнародовано, остается загадкой.

Мы напечатали эти заметки не только для тех, кто увлекается программой SHERLOCK. Они относятся ко всем, кто любит адвентюрные игры, но пока не может похвастаться особыми достижениями. Пусть действия Холмса послужат примером того, что можно требовать от хорошей программы этого жанра. А для тех, кто еще не начал работу с адвентюрными программами, мы только укажем, что все изложенное здесь представляет кратчайший путь к победе, но и на 10% не исчерпывает всех возможных вариантов развития событий.

Мы также рекомендуем начинающим изучить цикл статей "ADVENTURE LESSONS" в номерах ZX-РЕВЮ 1991 года.

НАШ КОНКУРС

В номере 1-е ZX-РЕВЮ за этот год, мы объявили конкурс по программе ELITE на самый рискованный маршрут.

Напомним УСЛОВИЯ: в одной из галактик надо облететь 20 планет, не побывав на одной планете дважды. Подсчет уровня опасности избранного маршрута производится по следующим критериям:

- анархическая планета - 5 очков;
- феодальная планета - 4 очка;
- любая другая - 1 очко.

Призы в конкурсе: пять первых мест получают ксерокопию повести DARK WHEEL на английском языке. Повесть написана по мотивам программы ELITE.

Как обычно, конкурс вызвал большой интерес среди наших читателей, по нему поступило около 100 писем с предложениями самого "крутого маршрута". Абсолютное большинство читателей пришли к выводу, что самым злачным местом во Вселенной является Галактика N4. Так получилось, что те читатели, которые пытались проложить маршрут по другим галактикам, не добрали очков и остались вне числа призеров.

Наилучший результат. достигнутый нашими пилотами - 100 баллов достигнут Ивановым А.И. из Читы. Мы, правда, должны сказать, что были еще такие же достижения, но получались они с нарушением правил. Например, когда пилот прокладывал маршрут по двум и более галактикам, пользуясь гипердрайвом. Такие решения мы не рассматривали.

На втором месте с результатом 92 балла оказалось сразу 12 человек.

Вот как выглядит наш Зал Славы:

Дзреев К.К. (Ростов на Дону)
Довженко В.П. (Киев)
Жаров Р.Н. (Херсон)
Игнатов А. (Новосибирск)
Коротков О.Е., Шилов А.В. (Ростов на Дону)
Минаков Р.С. (Калининград)
Минеев А.В. (Владивосток)
Мясников Г.Л. (Сыктывкар)
Радзевич А.А. (Нальчик)
Сергиенко Т.П. (Волжский, Волгоградской обл.)
Тошев В.В. (Норильск)
Хохлов Д.В. (Санкт-Петербург)

Нет числа разбитым пиратским кораблям, теперь они долго будут усеивать четвертую галактику, а перед нами новая проблема: как разделить 5 призов между 13 победителями (у нас просто больше нет ни одного лишнего экземпляра).

Мы приняли такое решение:

Иванову А. И. отправляем повесть "DARK WHEEL", путем жеребьевки она же отправляется Игнатову А., Короткову О.Е., Радзевичу А.А., Тошеву В.В.

Остальным же победителям мы в качестве завоеванного приза вышлем имеющиеся несколько экземпляров книги Яна Логана и Фрэнка О'Хары "Полный дисассемблер ПЗУ" тоже на английском языке. Эта книга является библией всякого настоящего "синклериста" и мы надеемся, что победители не будут огорчены произведенной заменой.

А что же касается самой повести "DARK WHEEL", то сотни наших читателей спрашивают, почему бы нам не перевести ее на русский язык и не опубликовать по частям в ZX-РЕВЮ? Действительно, такое решение выглядит наиболее рациональным и мы просто упустили его из виду, не ожидая что она вызовет массовый интерес. Так мы и сделаем в ближайших выпусках.

Содержание

К ЧИТАТЕЛЮ
ОБЩИЕ ВОПРОСЫ 1,45,87,89, 133,221,222, 248, 261
BETA-BASIC 3.0
Особенности версии 3
Команды..... 47, 91,135, 179
Функции..... 183, 223
Сообщения об ошибках. Коды ошибок 229
Дж Харднан, Э. Хьюзон. 40 ЛУЧШИХ ПРОЦЕДУР
Общие сведения о Бейсике и машинных кодах.....17
Внутренняя структура ZX-Spectrum. Карта памяти 16
Регистры Z-80..... 22
Загрузчик машинного кода MC-LOADER (программа)..... 24
Программы в машинных кодах Операции с экраном.... 26, 61
Процедуры обработки Бейсик-программ..... 105, 147
Форматы данных..... 166
Микайленко В.С. ЗАЩИТА ПРОГРАММ
Исключение возможности остановки программ.....9
Метолы защиты от просмотра..12
Управляющие коды..... 53
Методы защита от MERGE.... 59
Прочие приемы..... 97
Техника взлома программ:
Введение..... 100
Блокировка автозапуска. 104
Методика просмотра Бейсик-программ..... 144, 185
Изучение блоков в машинных кодах..... 187
Погоды известных взломщиков компьютерных программ..... 190
Методы защиты программ от копирования..... 235
РУСИФИКАЦИЯ ПРОГРАММ
Алексеев А. г. Русификация программы "MASTERFILE-09".. 29. 71
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД
Обработка ошибок в Бейсике ON ERROR GO TO(процедура) 113
Универсальный дебют.... 194
Заполнение REM... 194
Универсальное меню... 197
Создание и редактирование звуковых эффектов.... 241.
Ввод параметров при помощи оператора INPUT.... 245
ОШИБКИ ПЗУ
Обзор материалов зарубежной печати..... 209 238.
КАНАЛЫ И ПОТОКИ (Возвращаясь к напечатанному)
Пашорин В.И. Некоторые приемы защиты программ.....11
ГЛАВЫ ИЗ КНИГ
Элементарная графика..... 151
Векторная графика..... 204
Случайная графика..... 161
Криббедж..... 167
КАК ЭТО ДЕЛАЕТСЯ!
С. Тернер. Программа RANARAMA
Технология создания..... 249
Программа "SOUNDTESTER" для создания и редактирования звукового сопровождения игр... 254
СОВЕТЫ ЭКСПЕРТОВ
STALINGRAD, "CCS"..... 33

FLIGHT SIMULATION, "Psion".... 35
STAR RAIDERS 2. "ATARI"..... 36
THE TRAIN, "Accolade"..... 37
DEATH WISH 3 "Gremlin".....38
SHERLOCK 43. 129
PROFESSIONAL TENNIS "Dinamic".. 77
SNOOKER..... 77
QUAZATRON..... 79
CAPTAIN FIZZ "Psyclapse".... 82
ACADEMY "CRL GroUP",1986.(подробное описание).....125
BLINKY'S..... 211
OFERATIOH HORMUZ "Durell"..... 213
ACE "Cascade" 213
ACE-2 "Cascade"..... 214
TOMAHAWK "Digital Integr ".. 215
SKY RANGER "MicroSPhere"..... 216
TYPHOOH "Ocean"..... 216
TOP GUH "Ocean"..... 217
ATF "Digital integration" ...257
FLYER FOX "Bug Byte"..... 258
COBRA FORCE "Players" 256
THUNDER BLADE "U.S. Gold" ... 259
SANXION "Chaiamus"... 259
AIR WOLF "Elite"..... 260
1943 THE BATTLE OF MIDWAY.. 260
ПЕРЕПИСКА С ЧИТАТЕЛЯМИ (FORUM)
Вопросы совместимости
"Дубна-48" 119
"Ленинград-1" 42,120
"Пентагон 48"..121
Вопросы и проблемы, связанные с игрой ELITE ... 39, 84, 122, 131
Просьба о помощи (эксплуатация локальной сети 43
POKES и советы к играм IMPACT, SPECTRE OF BAGDAD ... 86
Вопросы циклической защиты программ....117
Адвентюрные игры. Программа HEAVY ON THE MAGIC..... 117
Советы и секреты (Пароли, POKES) 119
POKES..... 40, 46, 86, 119
Письмо читателя..... 119
МАЛЕНЬКИЕ ХИТРОСТИ
Удаление группы строк.... 203
Очистка памяти (экрана) ...208
МОНИТОР 48 . Новые возможности. Инструкция по пользованию.. 191
СПЕКТРУМ В ШКОЛЕ
Черкасский В. А. Бейсик программы "LISTNAME", REMONT"..... 2
Бейсик программа для проверки знаний на уроке истории.... 45
Бейсик-программа для урока географии 89
Экзаменующая программа .. 177
РЕКЛАМА
СИСТЕМА "РЕГИСТРАТУРА" 44
Комплекс "БАРЬЕР" 88
"DATA BASE PROCESSOR" 132
THE DARK WHEEL. Роберт Холдсток. Научно фантастическая повесть по мотивам программы ELITE ...175, 218, 262