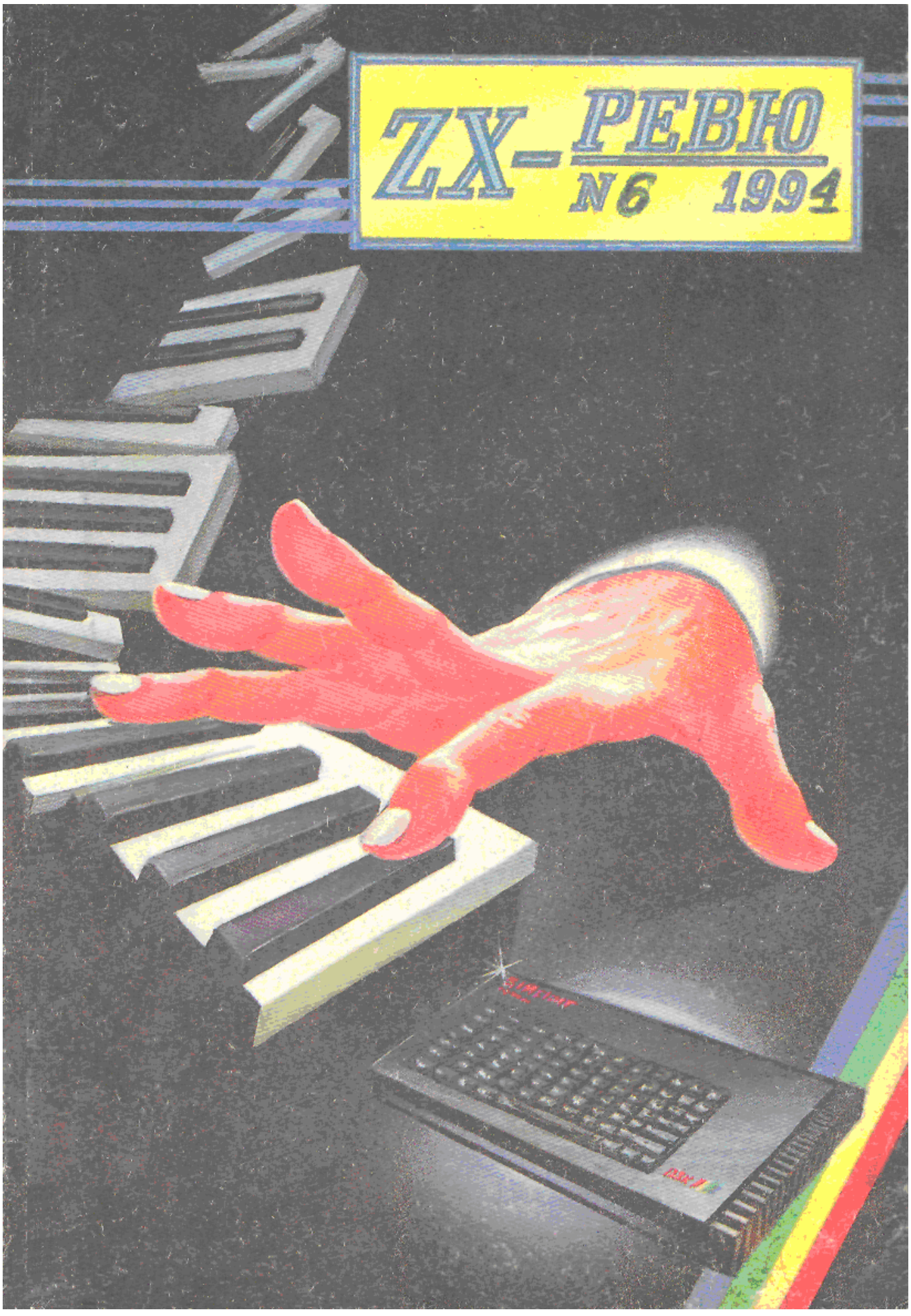
«ИНФОРКОМ»

PDF version by Deny (Денисенко Д.А.)
e-mail: DenyDA@mail.ru

ZX-РЕВЮ

№6 1994



В НОМЕРЕ:

КОМПЬЮТЕРЫ, КОТОРЫЕ МЫ ВЫБИРАЕМ		
От "Ленинграда" до "АТМ"	Вячеслав Толстой	2
Устранение несовместимости компьютера "Робик"	Александр Несвоваль	3
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД		
Генератор случайных чисел	Darky & Serge Sn.Damage Inc	5
ЧИТАТЕЛЬ - ЧИТАТЕЛЮ		
Имитатор "телевизора"	Патрахин В.В.	7
Экран высокого цветового разрешения	Александр Кунгуров	9
Несколько устройств для "Спекки"	Александр Кунгуров	17
Нестандартные загрузчики	Смоленцев А.В.	19
	Павел Бобровников	22
	Владимир Кашников	26
ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ		
Расширение памяти "Ленинграда-1" до 128K	Беззубцев А.С.	30
Некоторые подробности о мониторе TURBO-90	Игорь Назаров	31
ЭТЮДЫ		34
ФОРУМ		48
ПЕРЁКРЕСТОК		
Adventure Games		62
РОЛЕВЫЕ ИГРЫ		
Система правил AD&D (продолжение)	Сергей Симонович	71
Проект RPG	Сергей Симонович	81
СОВЕТЫ ЭКСПЕРТА		
"TAU CETI III"	Михаил Кретов, Артём Елисеев	86
ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ		
"OCEAN CONQUEROR"	Мажников Д.В.	93
"THE GREAT ESCAPE"	Олег Балуев	93
АВТОРСКАЯ ПРОГРАММА		94

От "Ленинграда" до "АТМ"

© Вячеслав Толстой, г. Петрозаводск, 1994.

У меня есть маленький "Ленинград" и уже третий из семейства "ATM-TURBO" - "TURBO 2+". О них я и хотел бы поведать. Надеюсь, моя информация окажется полезной.

В прошлом письме я сообщал, что на "TURBO-2, 2+" не идут программы FCboot, DCU версий 2.02 и выше. Я нашел причину - это программы, которые защищены и проверяют факт нажатия кнопки MAGIC. Они используют три байта из ПЗУ TR-DOS по адресам 02559-02561. В моём компьютере по этим адресам находится подпрограмма из программы обработки клавиатуры IBM. Чтобы эти программы работали (и многие другие), необходимо перепрограммировать ПЗУ TR-DOS.

Для версии BIOS 1.06 необходимо внести следующие изменения:

Ввести с адреса 02557:

```
#18, #03, #FF, #FF,
#FF, #C3, #33, #3C
```

С адреса 02565 до 02572 - `#FF`,

С адреса 15411 ввести:

```
#62, #A2, #5C, #38,
#0A, #00, #12, #00,
#00, #21, #00, #00,
#22, #FF, #81, #C9
```

Компьютер при инициализации проверит контрольную сумму ПЗУ TR-DOS, поэтому я изменил еще несколько байтов для восстановления общей суммы (обязательно) с адреса 04305: `#FF, #FF, #FF, #FF, #75`. Конечно, это по варварски, но мои познания в кодах не позволяют более детально осветить этот вопрос.

То же, для BIOS 1.07.00.

С адреса 02552:

```
#C3, #33, #3C, #FF,
#FF, #FF, #FF, #FF,
#FF, #FF, #FF, #FF
```

С адреса 04305:

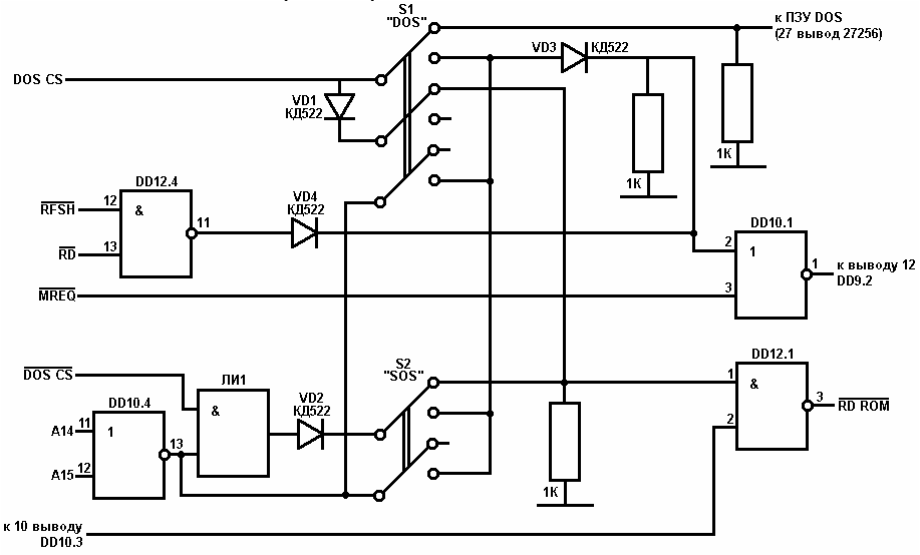
```
#FF, #FF, #51
```

С адреса 15411:

```
#41, #3E, #01, #17, #10, #FD,
#EE, #00, #32, #FA, #81, #C9
```

Наверное, я не нашел бы причину, если бы не мой маленький "Ленинград". Конечно, он у меня доработан и переработан по последнему слову. Так вот, последнее слово было такое. Вы много писали об использовании теневой области для загрузки в нее SOS. Я давно этим пользуюсь, но мне необходимо было исследовать различные версии TR-DOS (у меня "Ленинград", естественно, уже с дисководом), и я доработал схему следующим образом (см. схему).

За основу взята схема из ZX-РЕВЮ-94, № 2, стр. 52. Если у Вас в ПЗУ компьютера TR-DOS записана во вторую половину 27256, то схема подходит без изменений, Отключите сигнал "DOS CS" от 27 вывода и установите резистор 1 кОм на землю. Если TR-DOS в первой половине 27256, то вместо "DOS CS" подайте "инв. DOS CS" и резистор 1 к0м на +5В. Но на диод VD1 все равно придется подать "DOS CS".



DOS CS – сигнал выборки ПЗУ TR-DOS
$\overline{\text{DOS CS}}$ – инвертированный сигнал DOS CS

При помощи этой схемы я могу загрузить любую SOS и нормально работать с дисководом, а могу загрузить

любую DOS и так же работать без каких то проблем с SOS. После загрузки в теневую область новой OS, необходимо включить переключатель "SOS", а если Вы исследуете DOS, то - переключатель "DOS". Схема проста и не требует обращения к портам ввода/вывода.

Хотел бы еще остановиться на одном вопросе. Я хочу расширить свой маленький "Ленинград" до 128 (256) кб, но не на РУ5, а на РУ7. Схем расширения на микросхемах 565РУ5 было напечатано предостаточно, а вот на 565РУ7 - ни одной, хотя сами подумайте: поставить 8 шт. РУ7 - 256К или 16 шт. РУ5 - 128К. Мне кажется, РУ7 – дешевле. В связи с этим возникает вопрос об использовании памяти свыше 128К. На страницах РЕВЮ уже освещались эти вопросы. Но вот в № 4 РЕВЮ за этот год в статье А. Ларченко и С. Зонова на стр. 17 я прочитал, что специалисты фирмы "ISKRA-SOFT" обеспечили электронную поддержку расширенной памяти "Скорпиона". Теперь понятно - для того, чтобы мне использовать максимальную память с IS-DOS, мне необходимо память свыше 128К ориентировать на порт расширения #1FFD. Это порт расширения "Скорпиона". Очень бы хотелось узнать значение битов этого порта, и можно было бы браться за расширение маленького "Ленинграда". Это важно знать и программистам, которые хотят полностью использовать память "Скорпиона".

К вопросу о стандартизации. Если такая система, как IS-DOS взяла порт расширения свыше 128К #1FFD, то, может быть, всем на это и ориентироваться. Аппаратно любой порт организовать достаточно легко.

Если у кого-нибудь возникнут вопросы или проблемы, сообщаю свой телефон в Петрозаводске:

(814) 2-28-45

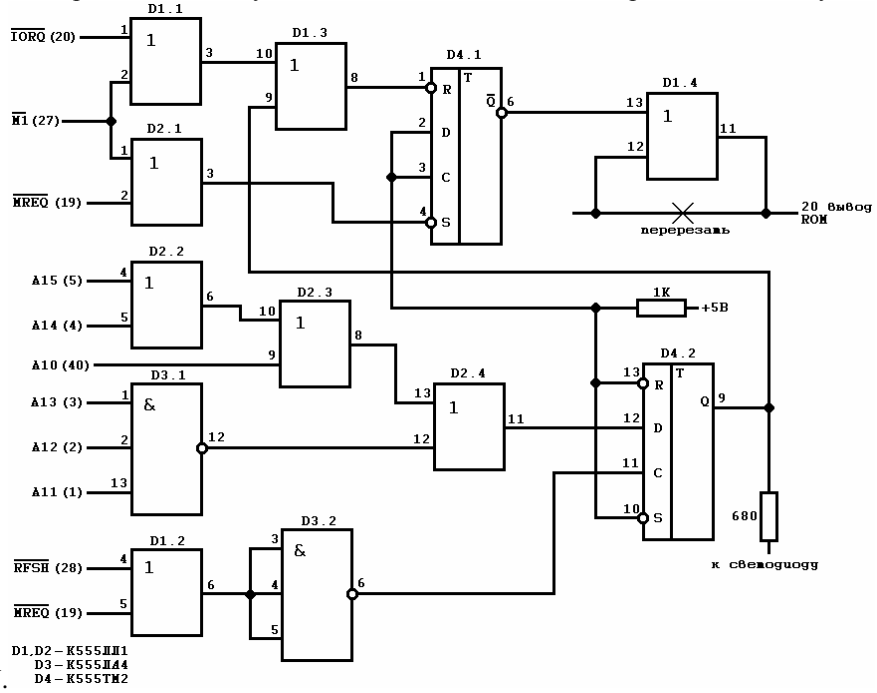
Всегда рад помочь.

Устранение несовместимости компьютера "Робик"

© Александр Несвоваль, г. Харьков.

На компьютерах типа "Робик" не работают некоторые игры, что вызывает у их владельцев чувство "ущербности" этого, очень неплохого, аппарата, Это происходит потому, что эти игры настраивают процессор на особый режим прерываний IM2, записывая при этом в регистр I числа #39, #3A или #3B. В "фирменной" ПЗУ 1982 г, с адреса #3900 по #3BFF находятся байты #FF. Поэтому, при любом считанном младшем байте, процессор в режиме IM2 при каждом прерывании будет переходить на адрес #FFFF. Обычно в этих играх по этому адресу записывают #18. А так как за адресом #FFFF следует адрес #0000, где находится в ПЗУ код #F3, то получается двухбайтовая команда JR #FFF4. При таком подходе экономится в памяти 257 байтов, потому, что любой программист добивается того, чтобы его программа в режиме IM2 работала на любом компьютере с любым считанным байтом. Для этого он участок памяти от #XX00 до #XXFF+1 заполняет одинаковым байтом, например, #FD. Тогда программа обработки прерываний будет работать с адреса #FDFD при любом считанном младшем байте. Хотя стандартом для ZX-Spectrum является считанный байт #FF, для чего шина данных через резисторы "подтягивается" к +5В.

Теперь немного о работе моей схемы (см. схему). На элементах D1.2, D2.2, D2.3, D3.1, D3.2 и D4.2 собрана аппаратная схема анализа содержимого регистра I. Если содержимое этого регистра находится в диапазоне #38-#3B, то это значит, что в момент прерывания происходит переход на адрес #FFFF. А как же этого добиться при заполненной области ПЗУ не #FF, а символьным набором, например, как в "Робике"? Очень просто: отключать ПЗУ в момент считывания вектора перехода и, затем, срезу же его включать. А так как шина данных "притянута" к +5В через резисторы, то мы и получим необходимое #FFFF даже при заполненном участке



ПЗУ.

Прежде чем приступить к переделке, помните, что компьютер достаточно сложная машина, которую

нетрудно вывести из строя не умелым вмешательством в схему и нелегко, потом отремонтировать. Поэтому не беритесь за доработку, если Вы не обладаете достаточным опытом работы с микропроцессорными системами, лучше обратитесь за помощью к более опытному человеку. Теперь немного о самой переделке. У микросхем D1-D4 отгибаются в сторону все выводы, кроме выводов питания 7 и 14 и устанавливаются они' поверх любых 14-вы водных микросхем 555 серии на плате компьютера. Выводы 7 и 14 дополнительных микросхем необходимо припаять к аналогичным выводам нижних микросхем, а отогнутые выводы укоротить бокорезами до 3-4 им. Монтаж ведется тонким проводом МГТФ. Микросхема D1.4 включается в разрыв провода, идущего к 20-ым выводам микросхем ПЗУ (в "Робике" это D42 и D43). Слева на схеме в скобках указаны номера выводов процессора Z80, к которым необходимо припаять эти выводы. По желанию, в схему можно установить светодиод ("минусом" к резистору, а "плюсом" к +5В) и вывести его на переднюю панель. Бели светодиод при запуске какой-нибудь игры горит, то это значит, что эта игра на не переделанном компьютере не работает, и вступила в работу предлагаемая схема.

Для примера я приведу лишь не большой перечень таких программ и задаваемое содержимое регистра I:

TOMAHAWK	#39	MIKIE	#39
ROBIN	#39	SMURF 1	#39
SUPER 3.5	#3B	GRYZOR 2	#3B
RAMBO	#3A	SPITFIRE	#3B
3D-LABYRINT	#39	TYPHOON	#39
ENTERPRI	#3A	WEC LE MANS	#3B
IMPACT	#39	ATARI POSITION	#3A
BOMB JACK	#39	SABOTEUR 2	#3B
SIGMA-7	#3B	ZERO MUSIC	#3B
GRYZOR 1	#3B	GRYZOR 3	#3B
CYCLONE	#3A	JUNGLE WARRIOR	#3B
COMET	#39	MATCH DAY 2	#3B
PLANET PATROL	#39	LOGO	#3B
SURV.	#39	PINBALL POWER	#3B

Генератор случайных чисел

© Darky & Serge Sh. Damage Inc, г. Нижний Новгород, 1994.

Мы решили поделиться крайне полезной, на наш взгляд, процедурой. Не секрет, что в программах, требующих многократных вызовов генератора случайных чисел, на это затрачивается львиная доля времени. Как правило, чтобы не изобретать велосипед, большинство программистов используют встроенную функцию ПЗУ, которая работает крайне медленно вследствие активного использования встроенного калькулятора. Мы также не считаем себя изобретателями велосипедов и решили просто "облегчить конструкцию".

Немного теории. Функция ПЗУ генерирует случайное число по следующей формуле: SEED=(((SEED+1)*75) MOD 65537)-1,
где SEED - это системная переменная Бейсика (23670/23671). Каждый раз при обращении к функции RND в нее заносится очередное, полученное по этой формуле, число. Сама же функция RND возвращает в Бейсик значение SEED/65536, что и дает число в диапазоне [0,1]. Посмотрим на числа, образующие формулу SEED: 1; 75, 65537… где здесь плавающая точка?!! Зачем здесь встроенный калькулятор? Об работку трехбайтовых чисел можно реализовать и на Ассемблере. Мы хотим предложить Вам вариант вычисления случайного числа по тому же алгоритму без использования "плавающей" арифметики (и калькулятора!).

Итак, процедура RND.

```
00010             ORG       40000
00020             ENT
00030             LD        HL,(SEED)
00040             INC       HL
00050             LD        D,0       ;Инкремент трехбайтового
00060             LD        A,H       ;числа, которое образуют
00070             OR        L         ;регистры D, H, L
00080             JR        NZ,NOCARRY ;
00090             INC       D
00100 NOCARRY     PUSH      DE        ;Сохранили новый SEED
00110             PUSH      HL
00120             SLA       L
00130             RL        H
00140             RL        D         ;Умножили на 2
00150             PUSH      DE        ;-Сохранили SEED*2
00160             PUSH      HL
00170             SLA       L
00180             RL        H
00190             RL        D         ; Умножили на 4
00200             SLA       L
00210             RL        H
00220             RL        D         ;Умножили на 8
00230             PUSH      DE        ;Сохранили SEED*8
00240             PUSH      HL
00250             SLA       L
00260             RL        H
00270             RL        D         ;Умножили на 16
00280             SLA       L
00290             RL        H
00300             RL        D         ;Умножили на 32
00310             SLA       L
00320             RL        H
00330             RL        D         ;Умножили на 64
00340             POP       BC
00350             POP       AF
00360             ADD       HL,BC
00370             ADC       A,D
00380             LD        D,A       ;SEED*64+SEED*8
00390             POP       BC
00400             POP       AF
00410             ADD       HL,BC
00420             ADC       A,D
00430             LD        D,A       ;SEED*64+SEED*8+SEED*2
00440             POP       BC
00450             POP       AF
```

```
00460             ADD       HL,BC
00470             ADC       A,D       ;SEED*64+SEED*8+SEED*2+SEED
;то есть SEED*75
00480             LD        C,A
00490             LD        B,0
00500             OR        A
00510             SBC       HL,BC
00520             ADC       HL,BC
00530             OR        A
00540             SBC       HL,BC     ;MOD 65367
00550             DEC       HL        ;Декремент SEED
00560             LD        (SEED),HL
00570             RET
00580 SEED        DEFW      0
Start of object:  #9C40 (40000)
Length:           #0055 (00085)
Executes:         #9C40 (40000)
```

Подпрограмма возвращает в регистр HL (и в пару ячеек SEED) псевдослучайное число в диапазоне [0,65535]. Делайте с ним, что хотите.

Мы пожертвовали объемом ради скорости работы, поэтому однотипные участки текста не были выделены в отдельные подпрограммы.

Шестнадцатеричный дамп процедура RND:

```
9C40 | 2A  93  9C  23  16  00  7C  B5 | 9F
9C48 | 20  01  14  D5  E5  CB  25  CB | 8E
9C50 | 14  CB  12  D5  E5  CB  25  CB | 52
9C58 | 14  CB  12  CB  25  CB  14  CB | 7F
9C60 | 12  D5  E5  CB  25  CB  14  CB | 62
9C68 | 12  CB  25  CB  14  CB  12  CB | 8D
9C70 | 25  CB  14  CB  12  C1  F1  09 | A8
9C78 | 8A  57  C1  F1  09  8A  57  C1 | 52
9C80 | F1  09  8A  4F  06  00  B7  ED | 99
9C88 | 42  ED  4A  B7  ED  42  2B  22 | D0
9C90 | 93  9C  C9  00  00  00  00  00 | 24
```

Мы вставили этот генератор в свои программы ("Минер", "Пасьянс") и результат нас поразил! Пришлось убирать из программ все надписи типа: "Please, wait…", так как их все равно никто не успел бы разглядеть.

Будем надеяться, что наша работа кому-то поможет.

Безусловно, поможет. Мы в этом уверены. Да вот, посмотрите, хотя бы, следующую статью…

Имитатор "телевизора"

© Патрахин В.В., г. Оленегорск, 1994.

Программа создает эффект "снега" при отсутствии сигнала на входе телевизора.

В программе используются переменные:

YX — двухбайтная величина, старший байт - строка, младший - столбец левого верхнего угла экрана имитируемого "'телевизора".

VERT — размер по вертикали (в знакоместах) экрана имитируемого телевизора.

HORIZ — то же по горизонтали.

```
00010             ORG       40000
00020             ENT
00030 YX          EQU       #0904
00040 VERT        EQU       10
00050 HORIZ       EQU       14
00060 START       LD        DE,0
00070             CALL      SNOW
00080             LD        DE,4000
00090             CALL      SNOW
00100             LD        DE,7000
00110             CALL      SNOW
00120             LD        DE,9000
00130             CALL      SNOW
00140             LD        A,127
00150             IN        A,(254)
00160             RRA
00170             JR        C,START
00180             RET
00190 SNOW        LD        HL,YX
00200             LD        A,L
00210             AND       #07
00220             RRCA
00230             RRCA
00240             RRCA
00250             ADD       A,H
00260             LD        H,L
00270             LD        L,A
00280             LD        A,H
00290             AND       #18
00300             OR        #40
00310             LD        H,A
00320             LD        B,8
00330 L4          PUSH      HL
00340             PUSH      BC
00350             LD        C,VERT
00360 L3          LD        B,HORIZ
00370             PUSH      DE
00380             POP       IX
00390             PUSH      HL
00400             LD        DE,#0020
00410             ADD       HL,DE
00420             EX        DE,HL
00430             POP       HL
00440             LD        A,D
00450             CP        H
00460             JR        Z,L1
00470             PUSH      HL
00480             EX        DE,HL
00490             LD        DE,#0700
00500             ADD       HL,DE
00510             EX        DE,HL
00520             POP       HL
00530 L1          PUSH      DE
00540             TUSH      IX
```

```
00550             POP       DE
00560 L2          LD        A,(DE)
00570             LD        (HL),A
00580             INC       HL
00590             INC       DE
00600             DJNZ      L2
00610             POP       HL
00620             DEC       C
00630             JR        NZ,L3
00640             POP       BC
00650             POP       HL
00660             INC       H
00670             DJNZ      L4
00680             RET
```

Эффект будет наиболее полным, если добавить шипение.

Комментарий ИНФОРКОМА.

Предложенная программа основана на том, что окно заданных размеров заполняется произвольной информацией, в качестве которой используется содержимое ПЗУ. Посмотрите: в строках 60, 80, 100, 120 в регистре DE задается некий произвольный адрес (в ПЗУ), а в строках 560-570 (в HL при этом находится адрес в дисплейном файле) происходит перенесение информации из ячейки ПЗУ на экран. За тем следует переход к следующей ячейке и повторение цикла (строки 580-600). Всего используется 4 повторения подпрограммы SNOW с разными адресами, благодаря этому и создается эффект "снега". Кстати сказать, в цикл переноса ин формации из ПЗУ в видео - ОЗУ удобно вставить звуковое сопровождение. Мы (по совету автора) сделали такое усовершенствование:

```
00572             AND    #10
00574             OR     #00
00576             OUT    (#FE),A
```

В строке 572 "выделяется" бит, соответствующий управлению динамикой (устраняются полосы на бордюре в такт шипению), в строке 574 задается цвет бордюра (для черного цвета эту строку можно исключить, но лучше - оставить для простоты внесения изменений в уже готовый блок; для белого бордюра здесь должно быть OR #07), в строке 576 полученное (случайное) значение аккумулятора выдается в порт звукового динамика, создавая шипение. Таким образом. Вы слышите, как "шипит" Ваше ПЗУ.

Параметры блока кодов со звуковым сопровождением:

```
Start of object:  #9C40 (40000)
Length:           #006A (00106)
Executes:         #9C40 (40000)
```

И его дамп:

9C40	11	00	00	CD	60	9C	11	A0	67
9C48	0F	CD	60	9C	11	58	1B	CD	0D
9C50	60	9C	11	28	23	CD	60	9C	0D
9C58	3E	7F	DB	FE	IF	38	E1	C9	8B
9C60	21	04	09	7D	E6	07	0F	0F	B2
9C68	0F	84	65	6F	7C	E6	18	F6	DB
9C70	40	67	06	08	E5	C5	0E	0A	83
9C78	06	0E	D5	DD	E1	E5	11	20	D1
9C80	00	19	EB	E1	7A	BC	28	08	67
9C88	E5	EB	11	00	07	19	EB	E1	F1
9C90	D5	DD	E5	D1	1A	77	E6	10	1B
9C98	F6	00	D3	FE	23	13	10	F4	35
9CA0	E1	0D	20	D4	C1	E1	24	10	F4
9CA8	CB	C9	00	00	00	00	00	00	D8

Тестируя программу, мы отмети ли, что четырех повторений цикла SNOW может быть и достаточно для простейшей имитации "снега", но для более реалистичной - явно не достаточно - слишком заметен полный цикл из 4-х повторений. В идеале было бы желательно организовать один замкнутый цикл с использованием в качестве генератора адреса для регистра DE - генератор случайных чисел. Если Вы уже прочитали предыдущую статью, то, конечно, подумали о том же самом, что и мы: интересно было бы узнать, как поведет себя предложенный генератор RND в процедуре "телевизора". Для этого необходимо внести следующие изменения:

```
00060 START       CALL   RND
00070             LD     DE,(SEED)
00080             LD     A,D
00096             AND    #3F
00100             LD     D,A
```

Строки 110-120 - удалите, а следом за процедурой SNOW расположите генератор случайных чисел из предыдущей статьи - подпрограмму RND. Результат её работы, находящийся в ячейке SEED, и будем использовать в

качестве адреса, задаваемого в DE. Это будет число от 0 до 65535. В этом случае, естественно, результат работы такого "телевизора" будет за висеть от содержимого памяти ПЗУ+ОЗУ Вашего компьютера, поэтому введено ограничение на старший байт адреса: строка 90 не дает выходить за пределы ПЗУ.

Когда мы запустили программу, то были просто "сражены наповал" реалистичностью эффекта: на экране нашего "Спектрума" светился (и шипел) настоящий маленький теле экран, время, от времени пытаясь настроиться на какую-то слабую станцию, но у него это никак не получалось. Такому поведению "телевизора" быстро нашлось объяснение: в ПЗУ 1982 года имеется свободная область, заполненная кодом #FF, и когда генератор случайных чисел попадал в этот диапазон, создавался эффект попыток "захвата" АПЧ слабого сигнала передаю щей станции. Эффект "захвата сигнала" почти пропадает при использовании ПЗУ "TURBO-90" (в котором нет свободного участка, но, за счет инкремента регистра DE в строке 590, в процесс вовлекается часть видео - ОЗУ) и совсем исчезнет, если изменить строку:

```
00090             AND    #1F
```

отрезав вероятность попадании о свободную область ПЗУ. Теперь видим полное отсутствие, какого бы то ни было сигнала на входе нашего "телевизора", но, все равно, очень реалистичный "снег" на мини-экране. Правда программа стала длиннее почти вдвое:

```
Start of object:  #9C40 (40000)
Length:           #OOB5 (00181)
Executes:         #9C40 (40000)
```

Однако если процедура RND будет также использоваться в Вашей программе и для других целей, то её применение заодно и в "телевизоре", наоборот, сэкономит память. А полученный эффект, безусловно, покорит тех, кто будет пользоваться Вашими программами.

В заключение, приводим полный дамп блока кодов имитатора телевизора вместе с генератором RND.

```
9C40 | CD  A0  9C  ED  5B  F3  9C  7A | 36
9C48 | E6  3F  57  CD  56  9C  3E  7F | DC
9C50 | DB  FE  1F  38  EB  C9  21  04 | F5
9C58 | 09  7D  E6  07  0F  0F  0F  84 | 18
9C60 | 65  6F  7C  E6  18  F6  40  67 | E7
9C68 | 06  08  E5  C5  0E  0A  06  0E | E8
9C70 | D5  DD  E1  E5  11  20  00  19 | CE
9C78 | EB  E1  7A  BC  28  08  E5  EB | 16
9C80 | 11  00  07  19  EB  E1  D5  DD | CB
9C88 | E5  D1  1A  77  23  13  E6  10 | 97
9C90 | F6  00  D3  FE  10  F4  E1  0D | E5
9C98 | 20  D4  C1  E1  24  10  CB  C9 | 92
9CA0 | 2A  F3  9C  23  16  00  7C  B5 | 5F
9CA8 | 20  01  14  D5  E5  CB  25  CB | EE
9CB0 | 14  CB  12  D5  E5  CB  25  CB | B2
9CB8 | 14  CB  12  CB  25  CB  14  CB | DF
9CC0 | 12  D5  E5  CB  25  CB  14  CB | C2
9CC8 | 12  CB  25  CB  14  CB  12  CB | ED
9CD0 | 25  CB  14  CB  12  C1  F1  09 | 08
9CD8 | 8A  57  C1  F1  09  8A  57  C1 | B2
9CE0 | F1  09  8A  4F  06  00  B7  ED | F9
9CE8 | 42  ED  4A  B7  ED  42  2B  22 | 30
9CF0 | F3  9C  C9  00  00  00  00  00 | E4
```

Экран высокого цветового разрешения

© Александр Кунгуров, п. Нижний Архыз, (КЧР), 1994.

В РЕВЮ-93 № 9-10 на стр.198 Киселев Д. рассматривает вопрос о возможности компьютера "Балтик" работать с экраном высокого раз решения по атрибутам. Думаю, что предложение использовать для этой цели экран по адресу 16384 - не состоятельно: этот экран не только "сорит" в буфере принтера, но и захватывает область системных переменных и еще свыше 4 Кбайт области Бейсика. Возможно, авторы "Балтика" полагали переписать операционную систему (ОС), сдвинув служебные области в более старшие адреса. Но это уже был бы не " Spectrum".

Другой вариант: если у "Балтика" вывод 3 микросхемы D17 оторвать от "земли" и подать на нее "1", как это делается в "Spectrum-128", то получится еще 4 экрана. Два из них - по адресам 32768 и 49152, с высоким разрешением по атрибутам. Но, используя их, мы тут же потеряем совместимость с другими типами "Spectrum" и работать с этими экранами придется из ассемблера. ОС их не поддерживает.

Предлагаю третий вариант: доработать "Балтик" согласно рис.1 (см. на следующей странице).

В этом случае графический экран всегда остается на своем стандартном месте, то есть по адресу 16384. А файл атрибутов программно или вручную может быть переключен со своего стандартного места - 22528 и размера - 768 байтов (далее "L-файл"), на адрес 55296 и размер 6144 ("H-файл"). При такой организации, программа должна иметь два файла атрибутов - L-файл и H-файл. Это обеспечивает совместимость по атрибутам низкого разрешения с

другими типами "Spectrum". Необходимость же иметь L-файл заставляет нас делать картинку в обычном графическом редакторе, а затем - дорабатывать L-файл до H-файла в специальном редакторе, например, в предлагаемом ниже "H-ART". Хотя, если эта система выживет, в будущем возможен и обратный вариант, то есть доработка H-файла до L-файла. Но для этого нужно еще написать редактор типа "ARTSTUDIO" для картинок с высоким раз решением.

Доработка по рис.1 рассчитана на "Балтик", с памятью, расширенной до 128 Кбайт. Для обычных 48К - компьютеров из схемы надо удалить D1'.1 и D3'.2, а D3'.1 заменить на ЛИ1. Серия микросхем может быть 155, 555, 531 и т.д. Программное включение H-файла из Бейсика: OUT 126,16. Включение L-файла: OUT 126,0.

Редактор высокого разрешения "H-ART" позволяет загружать стандартный экран, транслировать L-файл в H-файл, сохранять промежуточные версии H-файла в буфере. Работа в редакторе заключается в подведении курсора к нужному месту, заданию курсору требуемого цвета и впечатыванию его в H-файл.

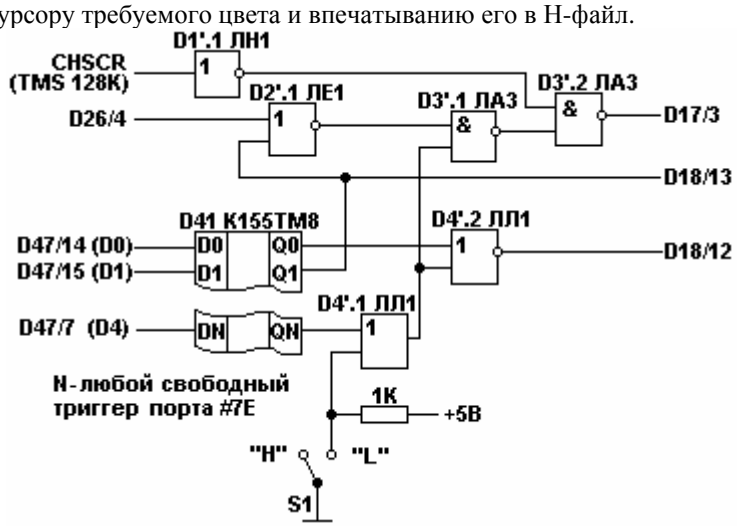


Рис. 1

Директивы "H-ART".

5, 6, 7, 8 - клавиши управления курсором.

0 - печать курсора.

In - установка цвета курсора (n-0…7) - INK.

Pn - установка цвета курсора (n=0…7) - PAPER.

B – вкл/выкл BRIGHT.

F - вкл/выкл FLASH.

S - переключение скорости курсора.

N - вкл/выкл непрерывной печати курсора.

C - графический экран с L-файлом копируется по адресу 48384. Затем, вместе с H-файлом сохраняется на магнитофоне в виде одного файла.

T0 - трансляция L-файла в H-файл.

D0 - пересылка H-файла в буфер.

I0 - пересылка содержимого буфера в H-файл.

Q0 - выход из режима редактирования H-файла.

В последних четырех директивах "0" вводится для утверждения директивы. Ввод любого другого символа отменяет директиву.

Таким образом, мы можем создавать картинки высокого разрешения по атрибутам или дорабатывать до высокого разрешения старые картинки. Но этого мало. Для полноты счастья, необходима возможность делать игры в высоком разрешении и, по возможности, в Бейсике. Бейсик не поддерживает H-файл, но зато он поддерживает L-файл. И, если делать обрамление в высоком разрешении, а игровое поле - в низком, то нам остается, не затрудняясь вычислениями, тупо перекачивать атрибуты из L-файла в H-файл. Либо, после каждого движения на экране, либо, если игра идет в постоянных атрибутах, после каждой смены картинки, обращаясь для этого к одной и той же подпрограмме. Для создания таких подпрограмм предназначена приведенная ниже программа "L-H SUBR".

"L-H SUBR" генерирует подпрограмму, призванную быстро транслировать атрибуты из L-файла в H-файл в окна заданных размеров. Генерация подпрограммы всегда производится по адресу 40000, но адреса в ней создаются для работы согласно запрашиваемому "L-H SUBR" реальному адресу загрузки. В подпрограмме может быть определено до 64-х окон с номерами 0…63. По окончании работы, "L-H SUBR" сообщает, кроме параметров подпрограммы, адрес SCR_NUM ячейки памяти, в которую записывается номер текущего окна. Из Бейсика менять окна можно с помощью оператора POKE SCR_NUM,n, где n - номер окна. Длина подпрограммы зависит от числа определенных окон, вернее от номера старшего определенного окна, и находится в пределах 136…388 байтов. Скорость работы подпрограммы почти линейно зависти от числа транслируемых знакомест экрана. При числе знакомест, равном 768 (весь экран, хотя это и бессмысленно) трансляция атрибутов происходит за 0.06 сек, что приемлемо при создании динамических игр. Запуск подпрограммы из Бейсика: RANDOMIZE USR ADR, где ADR - реальный адрес загрузки. К сведению: SCR_NUM=ADR+131. Окна, вернее их описания, занимая каждое - по 4 байта, располагаются с адреса ADR+132. В описании окна последовательно записываются параметры окна в знакоместах:

высота окна; ширина окна; координата Y верхнего левого угла окна, по отношению к левому верхнему углу экрана; координата X, определяемая так же (как в Бейсике).

Таким образом, эта система становится доступной и для тех, кто работает только на Бейсике. Но у владельцев "Балтиков" с памятью, расширенной до 128К, будут трудности. Ведь H-файл окажется на 7-й странице ОЗУ и придется из ассемблера заниматься переключением страниц. Мало этого - придется адаптировать программы, написанные для компьютеров с 48К памятью. Включение режима "48К" в этом случае проблемы не решает. Наиболее приемлемо - доработать компьютер согласно рис.2.

Такая доработка может быть полезна и для других типов компьютеров, поскольку, кроме H-файла, в текущей странице оказываются и экран - 1, и теневое ОЗУ. При этом экраном - 1 можно управлять через порт #7FFD. А, при использовании в качестве ОС "TURBO-90", появляется возможность, выходить из "зависших" программ с помощью кнопки сброса, не "вылетая" в ПЗУ 128К.

Теперь - листинги программ, о которых рассказывалось выше.

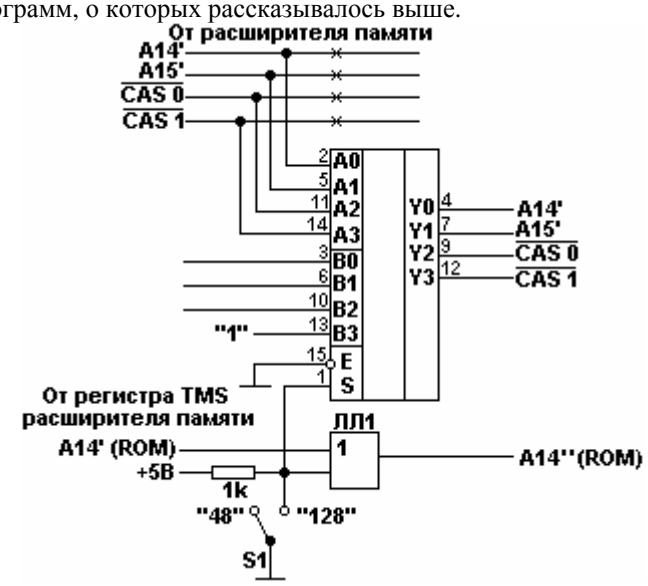


Рис. 2

Программа "H-ART".

```
10 REM H-ART Kungurov A.N.
15 LOAD "H-ART"CODE
20 BORDER 5: PAPER 5: INK 0: CLS: PAPER 6
30 PRINT AT 2,12; PAPER 6;"H -ART.": PLOT 95,151: DRAW 0,10: DRAW 64,0: DRAW 0,-10:
   DRAW -64, 0
40 PLOT 97,150: DRAW 63,0: DRAW 0,10
50 PLOT 99,149: DRAW 62,0: DRAW 0,10
60 PRINT AT 12,4;" 1 - LOAD L-screen "'AT 13,4;" 2 - LOAD L & H screens"
70 PRINT AT 14,4;" 3 - SAVE H-screen "'AT 15,4;" 4 - SAVE (L+H)-screen"
90 LET a$="                                ": PRINT AT 11,4;a$;AT 16,4 ;a$
100 PLOT 31,39: DRAW 0,49: DRAW 193,0: DRAW 0,-49: DRAW -193,0
110 PLOT 33,38: DRAW 192,0: DRAW 0,49
120 PLOT 35,37: DRAW 191,0: DRAW 0,49
130 LET a$=INKEY$: IF a$="" THEN GO TO 130
140 IF a$<"1" OR a$>"4" THEN GO TO 130
150 LET a=180+10*VAL a$: GO TO a
160 INPUT "L-screen name - ";b$: RETURN
170 INPUT "H-screen name - ";c$: RETURN
180 INPUT "(L+H)-screen name -";d$: RETURN
190 GO SUB 160: LOAD b$ CODE 16384: RANDOMIZE USR 61440: GO TO 20
200 GO SUB 160: GO SUB 170: LOAD b$ CODE 16384: LOAD c$ CODE 55296: RANDOMIZE USR
   61440: GO TO 20
210 GO SUB 170: SAVE c$ CODE 55296,6144: GO TO 20
220 GO SUB 180: SAVE d$ CODE 48384,13056: GO TO 20
```

Старт программы происходит с начальной Бейсик – строки. Прежде всего, после старта, загружается кодовый блок "H-ART" CODE 61440, 560. Вот его шестнадцатеричный дамп с контрольными суммами:

F000	F3	CD	D0	F1	CD	91	F1	32	F2
F008	35	F2	CD	FF	F1	3A	35	F2	3D
F010	FE	30	CA	0B	F1	FE	35	CA	F1
F018	13	F1	FE	36	CA	3B	F1	FE	34
F020	37	CA	4F	F1	FE	38	CA	2E	7F

F028	F1	CB	AF	FE	46	28	30	FE	1D
F030	42	28	47	FE	4E	CA	80	F1	58
F038	FE	50	28	4D	FE	49	28	57	B1
F040	FE	54	28	5B	FE	44	CA	C0	D1
F048	F0	FE	55	CA	E0	F0	FE	43	56
F050	CA	5F	F1	FE	53	CA	F1	F0	56
F058	FE	51	CA	03	F1	18	A5	3A	4C
F060	2F	F2	CB	7F	28	04	CB	BF	71
F068	18	02	CB	FF	32	2F	F2	CD	5C
F070	6E	F1	CD	D7	F1	CD	A7	F1	B9
F078	18	8A	3A	2F	F2	CB	77	28	CF
F080	04	CB	B7	18	E7	CB	F7	18	CF
F088	E3	CD	B1	F1	CB	21	CB	21	A2
F090	CB	21	E6	C7	B1	18	D5	CD	84
F098	B1	F1	E6	F8	B1	18	CD	CD	6B
F0A0	23	F2	11	00	58	21	00	D8	07
F0A8	3E	03	32	33	F2	CD	DC	F1	CA
F0B0	3E	07	84	67	3A	33	F2	3D	6C
F0B8	CA	04	F0	32	33	F2	18	ED	C2
F0C0	CD	6E	F1	CD	FA	F1	CD	23	84
F0C8	F2	21	00	D8	11	00	A0	01	55
F0D0	00	18	ED	B0	CD	6E	F1	CD	6E
F0D8	D7	F1	CD	A7	F1	C3	04	F0	AC
F0E0	CD	6E	F1	CD	FA	F1	CD	23	A4
F0E8	F2	21	00	A0	11	00	D8	18	8C
F0F0	DE	3A	00	F2	FE	60	28	07	77
F0F8	3E	60	32	00	F2	18	DB	3E	DB
F100	10	18	F7	CD	23	F2	CD	A7	66
F108	F1	FB	C9	3A	2F	F2	32	30	6B
F110	F2	18	C7	CD	09	F2	FE	00	98
F118	28	10	3D	CB	27	CB	27	CB	2D
F120	27	32	31	F2	CD	D0	F1	C3	DE
F128	04	F0	3E	F8	18	F3	CD	09	24
F130	F2	FE	1F	28	03	3C	18	E3	92
F138	AF	18	E6	CD	19	F2	FE	00	AC
F140	28	09	3A	32	F2	3D	32	32	61
F148	F2	18	D9	3E	BF	18	F7	CD	F5
F150	19	F2	FE	BF	28	06	3A	32	A3
F158	F2	3C	18	EA	AF	18	E7	21	48
F160	00	40	11	00	BD	01	00	1B	7B
F168	ED	B0	CD	A7	F1	C9	3A	32	90
F170	F2	47	3A	31	F2	4F	3E	BF	43
F178	CD	AC	22	3E	98	84	67	C9	8E
F180	3A	34	F2	FE	00	20	07	2F	25
F188	32	34	F2	C3	DA	F0	AF	18	25
F190	F7	21	3B	5C	CB	AE	3E	FF	E6
F198	32	00	5C	E5	FF	E1	AF	CB	56
F1A0	6E	28	F8	3A	08	5C	C9	21	A7
F1A8	6A	00	11	2C	01	CD	B5	03	C6
F1B0	C9	CD	91	F1	47	3A	35	F2	61
F1B8	B8	28	F6	78	0E	80	CD	01	53
F1C0	F2	D6	30	DA	04	F0	FE	08	7D
F1C8	D2	04	F0	4F	3A	2F	F2	C9	F2
F1D0	CD	6E	F1	7E	32	30	F2	3A	F9
F1D8	2F	F2	77	C9	06	00	22	36	88
F1E0	F2	0E	08	1A	77	24	0D	20	BB
F1E8	FB	13	2A	36	F2	23	05	20	81
F1F0	ED	C9	3A	34	F2	FE	00	28	1D
F1F8	01	C9	3A	30	F2	77	C9	0E	5D
F200	60	06	FF	10	FE	0D	20	F9	8B
F208	C9	CD	6E	F1	CD	F2	F1	3A	D9
F210	31	F2	CB	3F	CB	3F	CB	3F	43
F218	C9	CD	6E	F1	CD	F2	F1	3A	E9
F220	32	F2	C9	CD	B1	F1	AF	B9	D6

```
F228 | 20   01   C9   E1   C3   04   F0   A2 | 3E
```

Для тех, кто привык работать в ассемблере, приводим текст кодового блока программы "H-ART", набранный в формате "GENS-4'":

```
             ORG        #F000
             ENT        $
             DI
             CALL       PCUR
SCAN         CALL       KEY
             LD         (CK),A
             CALL       WAIT
             LD         A,(CK)
             CP         #30
             JP         Z,PRINT
             CP         #35
             JP         Z,LEFT
             CP         #36
             JP         Z,DOWN
             CP         #37
             JP         Z,UP
             CP         #38
             JP         Z,RIGHT
             RES        5,A
             CP         #46
             JR         Z,FLASH
             CP         #42
             JR         Z,BRIGHT
             CP         #4E
             JP         Z,NPR
             CP         #50
             JR         Z,PAPER
             CP         #49
             JR         Z,INK
             CP         #54
             JR         Z,TRANS
             CP         #44
             JP         Z,INBUF
             CP         #55
             JP         Z,OFFBUF
             CP         #43
             JP         Z,COPY
             CP         #53
             JP         Z,SPEED
             CP         #51
             JP         Z,QUIT
             JR         SCAN
FLASH        LD         A,(CUR)
             BIT        7,A
             JR         Z,FL
             RES        7,A
             JR         EFL
FL           SET        7,A
EFL          LD         (CUR),A
             CALL       SCUR
             CALL       PCN
             CALL       BEEP
             JR         SCAN
BRIGHT       LD         A,(CUR)
             BIT        6,A
             JR         Z,BR
             RES        6,A
             JR         EFL
BR           SET        6,A
             OR         EFL
PAPER        CALL       COL
```

```
            SLA         C
            SLA         C
            SLA         C
            AND         #C7
            OR          C
            JR          EFL
INK         CALL        COL
            AND         #F8
            OR          C
            JR          EFL
TRANS       CALL        CCOL
            LD          DE,#5800
            LD          HL,#D800
            LD          A,3
            LD          (NS),A
NSEC        CALL        TRS
            LD          A,7
            ADD         A,H
            LD          H,A
            LD          A,(NS)
            DEC         A
            JP          Z,SCAN
            LD          (NS),A
            JR          NSEC
INBUF       CALL        SCUR
            CALL        PBN
            CALL        CCOL
            LD          HL,#D800
            LD          DE,#A000
BUF         LD          BC,#1800
            LDIR
            CALL        SCUR
            CALL        PCN
BFA         CALL        BEEP
            JP          SCAN
OFFBUF      CALL        SCUR
            CALL        PBN
            CALL        CCOL
            LD          HL,#A000
            LD          DE,#D800
            JR          BUF
SPEED       LD          A,(WAIT+1)
            CP          #60
            JR          Z,SPD
            LD          A, #60
SD          LD          (WAIT+1),A
            JR          BFA
SPD         LD          A,#10
            JR          SD
QUIT        CALL        CCOL
            CALL        BEEP
El          RET
PRINT       LD          A,(CUR)
            LD          (BYTE),A
            JR          BFA
LEFT        CALL        WX
            CP          0
            JR          Z,RS
            DEC         A
RBYT        SLA         A
            SLA         A
            SLA         A
XBYT        LD          (XCUR),A
SBYT        CALL        PCUR
```

```
            JP          SCAN
RS          LD          A,#F8
            OR          XBYT
RIGHT       CALL        WX
            CP          #1F
            OR          Z,LS
            INC         A
            OR          RBYT
LS          XOR         A
            OR          XBYT
DOWN        CALL        WY
            CP          0
            OR          Z,TS
            LD          A,(YCUR)
            DEC         A
YBYT        LD          (YCUR),A
            OR          SBYT
TS          LD          A,#BF
            OR          YBYT
UP          CALL        WY
            CP          #BF
            OR          Z,LOS
            LD          A,(YCUR)
            INC         A
            OR          YBYT
LOS         XOR         A
            OR          YBYT
COPY        LD          HL,#4000
            LD          DE,#BD00
            LD          BC,#1B00
            LDIR
            CALL        BEEP
            RET
SCUR        LD          A,(YCUR)
            LD          B,A
            LD          A,(XCUR)
            LD          C,A
            LD          A,#BF
            CALL        #22AC
            LD          A,#98
            ADD         A,H
            LD          H,A
            RET
NPR         LD          A,(NP)
            CP          0
            OR          NZ,DPR
            CPL
EPR         LD          (NP),A
            JP          BFA
DPR         XOR         A
            OR          EPR
KEY         LD          HL,#5C3B
            RES         5,(HL)
            LD          A,#FF
            LD          (#5C00),A
KW          PUSH        HL
            RST         #38
            POP         HL
            XOR         A
            BIT         5,(HL)
            OR          Z,KW
            LD          A,(#5C08)
            RET
BEEP        LD          HL,#6A
```

```
            LD          DE,#12C
            CALL        #3B5
            RET
COL         CALL        KEY
            LD          B,A
            LD          A,(CK)
            CP          B
            OR          Z,COL
            LD          A,B
            LD          C,#80
            CALL        TIME
            SUB         #30
            CP          C,SCAN
            CP          8
            JP          NC,SCAN
            LD          C,A
            LD          A,(CUR)
            RET
PCUR        CALL        SCUR
            LD          A,(HL)
            LD          (BYTE),A
PCN         LD          A,(CUR)
            LD          (HL),A
            RET
TRS         LD          B,0
NSP         LD          (SCR),HL
            LD          C,8
            LD          A,(DE)
SPO         LD          (HL),A
            INC         H
            DEC         C
            OR          NZ,SPO
            INC         DE
            LD          HL,(SCR)
            INC         HL
            DEC         B
            OR          NZ,NSP
            RET
PBYT        LD          A,(NP)
            CP          0
            OR          Z,PBN
            RET
PBN         LD          A,(BYTE)
            LD          (HL),A
            RET
WAIT        LD          C,#60
TIME        LD          B,#FF
TM          DJNZ        TM
            DEC         C
            JR          NZ,TIME
            RET
WX          CALL        SCUR
            CALL        PBYT
            LD          A,(XCUR)
            SRL         A
            SRL         A
            SRL         A
            RET
WY          CALL        SCUR
            CALL        PBYT
            LD          A,(YOJR)
            RET
CCOL        CALL        COL
            XOR         A
```

```
            CP          C
            OR          NZ,ENTR
            RET
ENTR        POP         HL
            JP          SCAN
CUR         DEFB        #A2
BYTE        DEFB        0
XCUR        DEFB        0
YCUR        DEFB        0
NS          DEFB        0
NP          DEFB        0
CK          DEFB        0
SCR         DEFW        0
```

Программа "L-H SUBR":

```
10 REM L-H window subroutine          Kungurov A.N.
20 CLS : PRINT AT 2,4;  PAPER 5;  INK 2;" L-H window subroutine. "
30 LET a$="                            "
40 PRINT AT 1,4;  PAPER 5;a$;AT 3,4;a$
50 PLOT 31,143: DRAW 0,25: DRAW 193,0: DRAW 0,-25: DRAW -193,0
60 PLOT 33,142: DRAW 192,0: DRAW 0,25
70 PLOT 35,141: DRAW 191,0: DRAW 0,25
80 PLOT 37,140: DRAW 190,0: DRAW 0,25
90 INPUT "Real address - ";adr
100 RESTORE : LET k=0: PRINT AT 11,9;"Please wait..."
110 FOR n=1  TO 9
120 LET s=0
130 FOR m=1  TO 16
140 READ a: POKE 40000+k,a
150 LET k=k+1: LET s=s+a
160 NEXT m
170 READ a: IF s<>a THEN GO TO 200
180 NEXT n
190 GO TO 210
200 PRINT AT 11,7;  INK 2;  PAPER 4;  FLASH 1;"Error at line - ";200+n: BEEP .2,20:
    BEEP .2,20: BEEP .2,20: STOP
201 DATA 243,245,197,213,229,58,131,0,203,39,203,39,95,175,87,33,2190
202 DATA 132,0,25,17,127,0,1,4,0,237,176,42,129,0,235,205,1330
203 DATA 58,0,58,129,0,60,50,129,0,58,127,0,61,50,127,0,907
204 DATA 254,0,32,231,225,209,193,241,251,201,123,230,24,198,64,103,2579
205 DATA 62,152,132,103,123,230,7,15,15,15,130,111,34,125,0,123,1377
206 DATA 230,24,203,63,203,63,203,63,246,88,103,123,230,7,183,31,2063
207 DATA 31,31,31,130,111,235,58,128,0,79,42,125,0,229,6,8,1244
208 DATA 26,119,36,5,32,251,225,35,19,13,32,241,201,0,0,0,1235
209 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
210 FOR n=1  TO 12
220 READ a: LET a=a+40000: LET b=PEEK a: RANDOMIZE adr+b
230 LET c=PEEK 23670: LET d=PEEK 23671
240 POKE a,c: POKE a+1,d
250 NEXT n
260 DATA 6,16,20,28,32,35,39,42,46,77,103,107
270 BEEP 1,20: PRINT AT 11,9;"              ";AT 5,4;"Description of windows:"
280 LET mw=0
290 INPUT "Number of window - ";a
300 LET a=INT a: IF a<0 OR a>63 THEN GO TO 270
310 IF mw<a THEN LET mw=a
320 LET aw=40132+a*4
330 INPUT "X - ";x: LET x=INT x: IF x<0 OR x>30 THEN GO TO 330
340 INPUT "Y - ";y: LET y=INT y: IF y<0 OR y>22 THEN GO TO 340
350 INPUT "Width - ";sx: LET sx =ABS (INT sx): IF x+sx>31 THEN GO TO 350
360 INPUT "Height - ";sy: LET s y=ABS (INT sy): IF y+sy>23 THEN GO TO 360
370 POKE aw,sy: POKE aw+1,sx: POKE aw+2,y: POKE aw+3,x
380 PRINT INK 2;a; INK 0;"-";x;",";y;",";sx;"*";sy,
390 INPUT "Go on ? ";a$
400 IF a$="" OR a$="y" OR a$="Y" THEN GO TO 290
```

```
410 LET size=132+(mw+1)*4
420 SAVE "L-H SUbr."CODE 40000,size
430 CLS : PRINT "Address of subroutine - ";  INK 2;adr
440 PRINT '''"Address of byte SCR_NUM - ";  INK 2;adr+131
450 PRINT '''"Lenght of subroutine - ";  INK 2,size
460 PRINT '''"Number of screens - "; INK 2;mw+1
470 PRINT '''"Current screen - "; INK 2;PEEK 40131
```

Текст кодового блока, формируемого программой "L-H SUBR", набранный в формате "GENS-4":

```
        ORG     50000
        ENT     $
        DI
        PUSH    AF
        PUSH    BC
        PUSH    DE
        PUSH    HL
        LD      A,(ANW)
        SLA     A
        SLA     A
        LD      E,A
        XOR     A
        LD      D,A
        LD      HL,WIND
        ADD     HL,DE
        LD      DE,SYV
        LD      BC,4
        LDIR
NSTR    LD      HL,(YV)
        EX      DE,HL
        CALL    STR
        LD      A,(YV)
        INC     A
        LD      (YV),A
        LD      A,(SYV)
        DEC     A
        LD      (SYV),A
        CP      0
        JR      NZ,NSTR
        POP     HL
        POP     DE
        POP     BC
        POP     AF
        EI
        RET
STR     LD      A,E
        AND     #18
        ADD     A,#40
        LD      H,A
        LD      A,#98
        ADD     A,H
        LD      H,A
        LD      A,E
        AND     7
        RRCA
        RRCA
        RRCA
        ADD     A,D
        LD      L,A
        LD      (HADR),HL
        LD      A,E
        AND     #18
        SRL     A
        SRL     A
        SRL     A
        OR      #58
```

```
            LD          H,A
            LD          A,E
            AND         7
            OR          A
            RRA
            RRA
            RRA
            RRA
            ADD         A,D
            LD          L,A
            EX          DE,HL
            LD          A,(SXV)
            LD          c,A
            LD          HL,(HADR)
NPL         PUSH        HL
            LD          B,8
            LD          A,(DE)
PL          LD          (HL),A
            INC         H
            DEC         B
            JR          NZ,PL
            POP         HL
            INC         HL
            INC         DE
            DEC         C
            JR          NZ,NPL
            RET
HADR        DEFW        0
SYV         DEFB        0
SXV         DEFB        0
YV          DEFB        0
XV          DEFB        0
ANW         DEFB        0
WIND        DEFB        0
```

Несколько устройств для "Спекки"

© Кунгуров А.Н., п. Нижний Архыз, (КЧР), 1994.

Часто встречаются программы, рассчитанные на мастеров спорта по настольному теннису или на граждан своей страны, владеющих скорочтением. Не во всех программах предусмотрен режим ПАУЗА. Преодолеть эти трудности можно с помощью замедлителя темпа, позволяющего плавно регулировать скорость работы процессора вплоть до полной остановки. Схема, представлена на рис.1. Обозначение на этой и остальных схемах соответствует компьютеру "Балтик". Кроме того, для "Балтика" не требуется устанавливать VD2' и R4' и разрывать цепь "WAIT".

На схеме (рис.1) на D1.1' собран внешний генератор, так как использовать какую-либо частоту с платы компьютера нельзя из-за её синхронности с сигналом INT. Номинал R2' подбирается таким, что бы при положении R6' в состоянии максимального сопротивления, процессор останавливался полностью.

Настройку замедлителя темпа можно выполнить с помощью программы:

```
10 FOR a=1 TO 2
20 PRINT AT 10,0;
30 FOR b=0 TO 32
40 PRINT PAPER a;" ";
50 NEXT b
60 NEXT a
70 GO TO 10
```

Конечно, могут возникнуть сом нения в этичности пропагандирования такого устройства, поэтому, для тех, кто задается этим вопросом, предусмотрен выключатель S1.

Теперь относительно реализации порта #FF. Вообще говоря, информация, извлекаемая из этого порта - бессмысленна. Поэтому, не всё ли равно, откуда её брать? И вовсе не обязательно вводить в компьютер "мохнатого паука" и перегружать бедную шину данных. В схеме на рис.2 роль порта #FF выполняет порт #31 (KEMPSTON-джойстика).

Необходимо разорвать цепи, идущие к выводам 5, 8 и 9 внешнего интерфейса на К580ВВ55А и включить в разрывы эту схему. В качестве Fстроб. можно использовать кадровый синхроимпульс КСИ, либо частоту, взятую из замедлителя темпа. (Dl':13). При отсутствии сигнала "IN#FF", можно использовать простейший дешифратор,

например, изображенный на рис.3.

Хотя этот порт #FF и "неправильный", программы RENEGADE и SHORT CIRCUIT им вполне удовлетворяются. Кроме того, его можно использовать как генератор случайных чисел, если игнорировать нулевые значения переменной a (см. подпрограмму RND ниже).

Подпрограмма случайных чисел для порта #FF. Результат - в переменной a. Нулевые значения стоит игнорировать, поскольку, из-за ширины Fстроб. вероятность их по явления выше, чем других значений.

```
9910 REM Subroutine RND
9920 LET A=0
9930 LET b=IN 255: IF b=255 THEN LET a=a+l: GO TO 9930
9940 RETURN
```

Теперь о музыкальном процессоре. Во всех встреченных мной публикациях о подключении музыкального процессора AY-3-8910(12) вопрос о тактовой частоте либо обходится, либо утверждается, что она должна быть 1,75 МГц. Наконец, в книге Стивена Виккерса "Дополнение к книге Программирование на языке Бейсик", под редакцией Робина Бредбира . The Leagrave Press LTD, Luton and London, Great Britain Third Edition 1986, перевод 1989 г., удалось обнаружить, что в стандартной Spectrum частота тактирования AY-3-8910(12) равна 1,7734 МГц. Точность при этом 0,01%. Очевидно, что при тактовой частоте 1,75 МГц, точность составит 1,32%. В компьютерах с кварцем 16МГц делением на 9 можно получить частоту 1,778МГц с точностью тактирования 0,25%. В ZX-FORUM на стр. 146 А. Гармаисанов пишет: "… и звук хороший…". Выходит, бывает и плохой? Действительно, бывает. И еще, какой плохой! Дело в том, что AY-3-8910(12) сбивается, если во время обращения к нему компьютера приходят тактовые импульсы. Поэтому тактовая частота должна быть либо синхронизирована с обращениями компьютера, либо запрещена во время них. Последнее и реализовано на рис. 4.

Вместо делителя D2 '", D3 '" можно подключить любой внешний генератор. На качестве музыки та кое обращение с тактовой частотой практически не сказывается, поскольку AY-3-8910 (12) работает по фронту тактовой частоты и ему все равно, что приходит на вход CLK -меандр или "иголка".

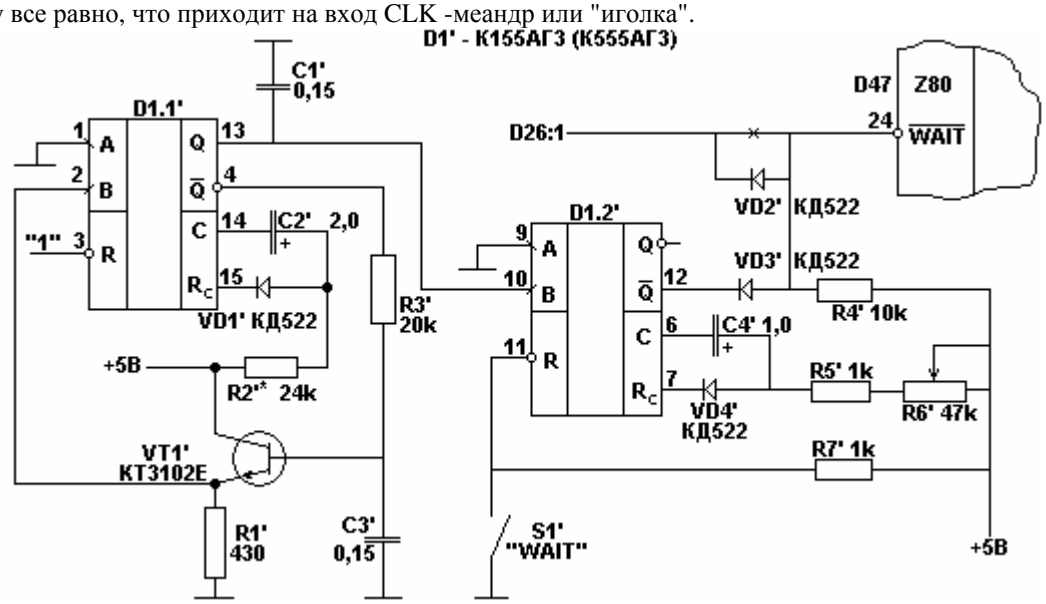


Рис. 1

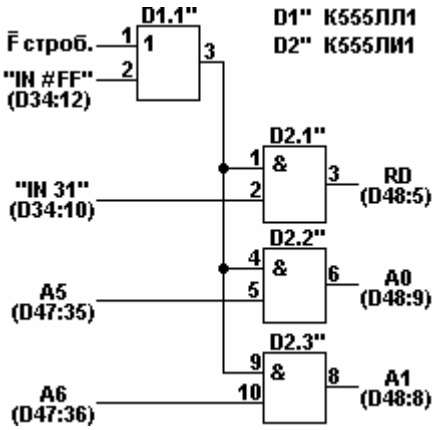


Рис. 2

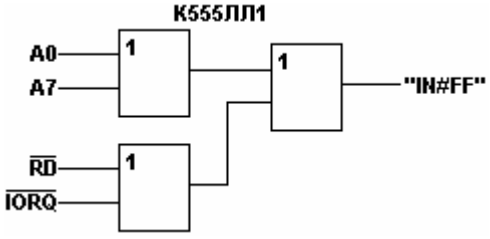


Рис. 3

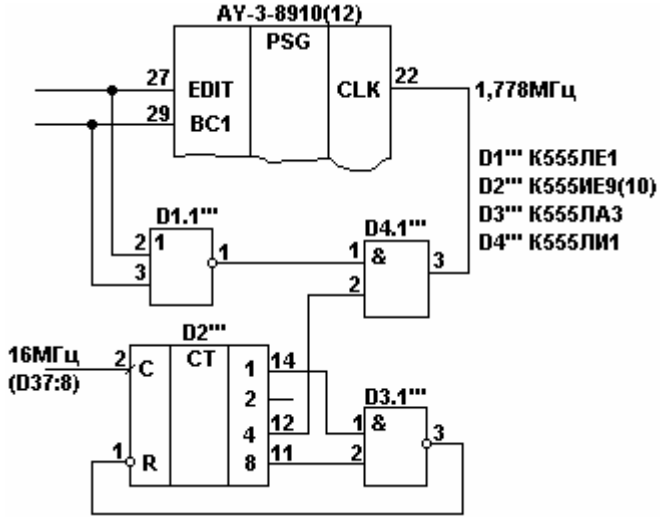


Рис. 4

Нестандартные загрузчики

Многие читатели обращаются к пак с просьбой опубликовать ин формацию о нестандартной загрузчике, в которой загрузка заставки на экран происходит по знакоместам а любой предварительно задан ной последовательности: при этом по экрану как бы "бегает" курсор, который прорисовывает картинку.

Сегодня мы располагаем такими материалами и можем поделиться ими с нашими читателями в этой подборке.

Загрузка экрана "бегающим курсором"

© Смоленцев А.В., г. Белебей, Башкирия, 1994.

Идея этого способа загрузки взята мною из программы N.O.M.A.D. фирмы OCEAN, однако загрузчик эк рана я подвергнул существенным изменениям в лучшую, я надеюсь, сторону.

Для начала о самом экранном файле нового формата. Данные о загружаемой картинке хранятся в следующем формате:

Байт 0 - младший байт адреса верхней пиксельной линии загружаемого знакоместа.

Байт 1 - старший байт адреса верхней пиксельной линии загружаемого знакоместа.

Байт 2-9 - последовательно восемь линий знакоместа сверху вниз.

Байт 10 - атрибут данного знакоместа.

Далее следует информация о другом знакоместе в том же порядке.

Таким образом, с помощью специального загрузчика можно загружать знакоместа экрана в произвольном порядке, причем в самом загрузчике отсутствуют какие-либо параметры загружаемого экранного файла, то есть загрузчик является полностью универсальным.

При таком способе загрузки ин формация о каждом знакоместе занимает 11 байтов и для полностью заполненного экрана файл будет длиной 11*32*24=8448 байтов. Очевидно, что данный способ загрузки может выиграть в скорости загрузки, если на картинке есть какие-то "не закрашенные" области.

Теперь листинг нестандартного загрузчика на Ассемблере (опять-таки с адреса 40000 только для примера).

```
00010          ORG       40000
00020          ENT
00030          CALL      BEGIN
00040          EI
00050          RET
00060 BEGIN    DI
00070          LD        A,#0F
00080          OUT       (#FE),A
00090          IN        A,(#FE)
00100          RRA
00110          AND       #20
00120          OR        #02
00130          LD        C,A
00140          CP        A
00150 BREAK    RET       NZ
00160 START    CALL      EDGE1
00170          JR        NC,BREAK
```

```
00180          LD        HL,#0415
00190 WAIT     DJNZ      WAIT
00200          DEC       HL
00210          LD        A,H
00220          OR        L
00230          JR        NZ,WAIT
00240          CALL      EDGE2
00250          JR        NC,BREAK
00260 LEADER   LD        B,#9C
00270          CALL      EDGE2
00280          JR        NC,BREAK
00290          LD        A,#C6
00300          CP        B
00310          JR        NC,START
00320          INC       H
00330          JR        NZ,LEADER
00340 SYNC     LD        B,#C9
00350          CALL      EDGE1
00360          JR        NC,BREAK
00370          LD        A,B
00380          CP        #D4
00390          JR        NC,SYNC
00400          CALL      EDGE1
00410          RET       NC
00420          LD        A,C
00430          XOR       #03
00440          LD        C,A
00450          LD        B,#B0
00460          CALL      MARKER
00470 LOOP     CALL      PLACE
00480          RET       NC
00490          JR        LOOP
00500 PLACE    LD        B,#B0
00510          CALL      MARKER
00520          LD        L,E
00530          LD        B,#B0
00540          CALL      MARKER
00550          RET       NC
00560          LD        H,E
00570          LD        D,#08
00580          LD        B,#B0
00590 GRAF     CALL      MARKER
00600          RET       NC
00610          LD        (HL),E
00620          INC       H
00630          DEC       D
00640          LD        B,#B1
00650          JR        NZ,GRAF
00660          LD        A,H
00670          SUB       #08
00680          RRA
00690          RRA
00700          RRA
00710          AND       #03
00720          OR        #58
00730          LD        H,A
00740          LD        B,#B2
00750          CALL      MARKER
00760          LD        (HL),E
00770          RET
00780 MARKER   LD        E,#01
00790 LD8BIT   CALL      EDGE2
00800          RET       NC
00810          LD        A,#CB
```

```
00820          CP        B
00830          RL        E
00840          LD        B,#B0
00850          JR        NC,LD8BIT
00860          RET
00870 EDGE2    CALL      EDGE1
00880          RET       NC
00890 EDGE1    LD        A,#16
00900 DELAY    DEC       A
00910          JR        NZ,DELAY
00920          AND       A
00930 SAMPLE   INC       B
00940          RET       Z
00950          LD        A,#7F
00960          IN        A,(#FE)
00970          RRA
00980          RET       NC
00990          XOR       C
01000          AND       #20
01010          JR        Z,SAMPLE
01020          LD        A,C
01030          CPL
01040          LD        C,A
01050          AND       #07
01060          OR        #08
01070          OUT       (#FE),A
01080          SCF
01090          RET
Start of object:  #9C40 (40000)
Length:           #OOB6 (00182)
Executes:         #9C40 (40000)
```

Большая, часть программы соответствует стандартной процедуре LD_BYTES, которая находится в ПЗУ с адреса #0556 (01366).

Для более глубокого понимания процессов, происходящих при загрузке блока кодов с магнитофона, мы можем посоветовать взять (или заказать у нас) подшивку ZX-РЕВЮ) за 1991. Там, в № 4-5 на стр. 71-79, подробно разобраны процедуры работы с магнитофоном. Загляните также и в подшивку ZX-РЕВЮ за 1993 год - на стр. 13-20 Вы найдете подробное описание стандартной процедура загрузки.

Используемые регистры: A,B,C - аналогично процедуре LD_BYTES.

D - счетчик цикла для загрузки 8 пиксельных линий знакоместа.

E - используется вместо регистра L в LD_BYTES, в нем "собирается" по битам принимаемый с ленты байт.

HL - хранение адреса в области экрана, куда будут перебрасываться данные из регистра E.

В программу можно внести несколько изменений по своему усмотрению. Если удалить строку 980, программа перестанет реагировать на клавишу BREAK. Если Вы не хотите, чтобы на бордюре при загрузке мелькали полосы, нужно уда лить строки 70, 80, 120, 420-440, 1050-1070.

В приведенной программе отсутствует контроль правильности считывания, так что если Вы попробуете загрузить этим загрузчиком не заранее подготовленный экранный файл, то результат будет непредсказуемый - как правило это приводит к зависанию системы, так как загружается неизвестно что, и неизвестно куда.

Шестнадцатеричный дамп загрузчика с контрольными суммами:

```
9C40 | CD  45  9C  FB  C9  F3  3E  0F | 8E
9C48 | D3  FE  DB  FE  1F  E6  20  F6 | A9
9C50 | 02  4F  BF  C0  CD  D8  9C  30 | 2D
9C58 | FA  21  15  04  10  FE  2B  7C | DD
9C60 | B5  20  F9  CD  D4  9C  30  EB | 22
9C68 | 06  9C  CD  D4  9C  30  E4  3E | 35
9C70 | C6  B8  30  E0  24  20  F1  06 | D5
9C78 | C9  CD  D8  9C  30  D5  78  FE | 99
9C80 | D4  30  F4  CD  D8  9C  D0  79 | 9E
9C88 | EE  03  4F  06  B0  CD  C4  9C | 47
9C90 | CD  96  9C  D0  18  FA  06  B0 | C3
9C98 | CD  C4  9C  6B  06  B0  CD  C4 | 13
9CA0 | 9C  D0  63  16  08  06  B0  CD | AC
9CA8 | C4  9C  D0  73  24  15  06  B1 | D7
9CB0 | 20  F5  7C  D6  08  1F  1F  1F | 18
```

```
9CB8 | E6  03  F6  58  67  06  B2  CD | 77
9CC0 | C4  9C  73  C9  1E  01  CD  D4 | B8
9CC8 | 9C  D0  3E  CB  B8  CB  13  06 | 75
9CD0 | B0  30  F3  C9  CD  D8  9C  D0 | 19
9CD8 | 3E  16  3D  20  FD  A7  04  C8 | 95
9CE0 | 3E  7F  DB  FE  1F  D0  A9  E6 | 90
9CE8 | 20  28  F3  79  2F  4F  E6  07 | A3
9CF0 | F6  08  D3  FE  37  C9  00  00 | 5B
```

Итак, предположим, что Вы набрали данный загрузчик, ассемблировали его и выгрузили на ленту. Тут появляется вторая проблема: а что этим загрузчиком загружать? Ведь ему требуется специальный блок кодов на ленте с закодированной информацией о порядке загрузки и содержимом знакомест экрана, эту проблему поможет успешно решить небольшая Бейсик-программа, которая из стандартного экрана, не без Вашего участия, сформирует закодированный блок. Эта скромная программа, приведенная ниже, не слишком удобна в управлении и имеет минимальный набор сервисных возможностей. Однако их вполне хватит, чтобы отладить загрузчик и закодировать один-другой экран. А дальше можно расширить её в соответствии со своими желаниями и возможностями.

Конечно, можно было бы сразу делать кодирующую программу в кодах и с шикарным сервисом, но вся прелесть Бейсика, несмотря на его "неповоротливость", заключается в том, что набрать, а тем более, отладить килобайт Бейсика гораздо легче и быстрее, чем 200 байтов машинного кода. А когда в Бейсике полностью отмакетирована и выяснена логика работы, можно заняться и машинным кодом.

```
10 PAPER 0: INK 7: BORDER 0: CLEAR 39999
20 LET a=22528: LET b=55000
30 LOAD "" SCREEN$
40 LET a$=INKEY$
50 LET a=a-(a$="6" AND a>22528)+(a$="7" AND a<23295)-32*(a$="9" AND
   a>22559)+32*(a$="8" AND a< 23264)
60 LET at=PEEK a: POKE a,0: PAUSE 5: POKE a,127: PAUSE 5: POKE a,at
70 IF a$="0" THEN GO TO 200
80 IF a$=" " THEN LET b=b-11: POKE a,at: BEEP 1,0
90 IF a$="1" THEN GO TO 300
100 GO TO 40
200 BEEP .1,30: LET ad=a-2560-1792*(a<23040)-1792*(a<22784)
210 LET ah=INT (ad/256): LET al=ad-256*ah
220 POKE b,al: POKE (b+l),ah
230 FOR n=0 TO 7: POKE (b+n+2), PEEK (ad+256*n): NEXT n
240 POKE b+10,at: LET b=b+11: LET at=at
250 POKE a,0: GO TO 40
300 INPUT "Filename : ";a$
310 SAVE a$ CODE 55000,(b-55000)
```

Назначение переменных в программе:

a - адрес атрибута текущего знакоместа;
b - первый свободный адрес создаваемого файла;
at - значение атрибута текущего знакоместа;
ad - адрес первой линии текущего знакоместа;
al, ah - младший и старший байты адреса верхней пиксельной линии знакоместа;
atl - значение атрибута текущего знакоместа для временного хранения.

Назначение строк программы:

40-100 - определение нажатой клавиши;
50 - пересчет адреса атрибута знакоместа в зависимости от перемещения курсора;
60 - печать курсора в текущем знакоместе;
80 - реакция на нажатие "пробела";
200-250 - пересылка данных из текущего знакоместа в создаваемый файл в нужном формате.

После запуска с начальной строки программа ждет первый, встретившийся на ленте файл? и попытается загрузить его в экран. После успешной загрузки картинки в левом верхнем углу появится курсор, который мигает то белым, то черным, то - цветом данного знакоместа. Курсор перемещается клавишами SINCLAIR-джойстика. При нажатии на "огонь" (при этом раздается короткий звуковой сигнал), содержимое знакоместа экрана запишется в буфер, а данное знакоместо станет черным. Так Вам предлагается последовательно закодировать весь экран. Последнюю запись можно аннулировать нажатием SPACE. Нажатием "1" можно выгрузить созданный файл на ленту. Загрузка экрана будет происходить в том порядке, в каком Вы производили запись знакомест в буфер.

Как уже говорилось, программа предназначена для отработки алгоритма, поэтому она имеет некоторые ограничения.

1. Если нажать "огонь" два раза на одном знакоместе, при загрузке здесь будет черный квадрат.

2. Если нажать на "пробел" два раза подряд. Вы отмените две последние записи, но на экране ничего не заметите.

3. Если Вы хотите отменить последнюю запись, установите курсор на это знакоместо, иначе будут изменены атрибуты не там, где надо.

4. Имя файла, естественно, не должно быть более 10 символов; при записи готового файла заголовок нужно пропустить.

В заключение хочу добавить, что загрузчик запускается с начального адреса и является неперемещаемой программой, т.е. должен работать только в тех адресах, куда был ассемблирован. Например, если Вы в 10 строке листинга 1 поставите:

```
00010          ORG       65000
```

то запускать загрузчик нужно следующим образом:

```
10 CLEAR 64999: LOAD "loader" CODE: RANDOMIZE USR 65000
```

А теперь предлагаем небольшое, но очень "емкое" письмо другого нашего корреспондента на ту же тему. Обратите на него, пожалуйста, вникание - едва ли Вы встретите загрузчик короче!

© Павел Бобровников, г. Челябинск, 1994.

Мое письмо я в большей степени посвящаю Спектруму и магнитофону (я думаю, меня поймут: мы все начинали с него). Привожу программу, предназначенную для загрузки кодовых блоков при помощи "бегающего курсора" в произвольном порядке, в зависимости от кодирования картинки.

```
00010          ORG       40000
00020          ENT
00030          LD        IX,M2+2
00040          LD        DE,2
00050          XOR       A
00060          SUB       1
00070          EX        AF,AF'
00080          DI
00090          CALL      #0562
00100 M2       LD        IX,0
00110          EXX
00120          LD        B,8
00130 M1       EXX
00140          INC       E
00150          CALL      #05A9
00160          DEC       IX
00170          DEFB      #DD
00180          DEFB      #24
00190          EXX
00200          DJNZ      M1
00210          DEFB      #DD
00220          DEFB      #7C
00230          RRCA
00240          RRCA
00250          RRCA
00260          DEC       A
00270          AND       #03
00280          OR        #58
00290          DEFB      #DD
00300          DEFB      #67
00310          EXX
00320          INC       E
00330          CALL      #05A9
00340          LD        IX,M2+2
00350          LD        E,2
00360          CALL      #05A9
00370          DEC       E
00380          INC       E
00390          JR        Z,M2
00400          LD        A,(M2+2)
00410          CP        1
00420          JP        #053F
Start of object:  #9C40 (40000)
Length:           #0048 (00072)
Executes:         #9C40 (40000)
```

Пользователю требуется только набрать этот кусок текста в ассемблере и запустить его с того адреса, куда он ассемблировался. Еще замечу, чтобы загрузка выглядела солидней, замените в строке 90 CALL #0562 на CALL #056A

- это вызовет изменение цвета полос.

Комментарий ИНФОРКОМА.

Не скроем, длина загрузчика нас просто покорила. Однако прежде, чем давать комментарии, проведем некоторые изменения. Обращаем Ваше внимание на строки 170-180, 210-220, 290-300. Что бы это могло означать? Да ни что иное, как недокументированные команды с половинками индексного регистра IX! В свое время Игорь Комаров из Иркутска (см. ZX-PEBIO-94, № 2, стр. 56-57) предлагал способ, как включать в ассемблерный текст такие команды. Воспользуемся этим приемом. Заодно сделаем замену цвета бордюрных полос, предложенную автором письма. Кстати, и без того короткую программу можно сократить еще на пару байтов, если вместо использования альтернативного набора регистров, сохранять данные B на стеке. Изменению подверглись строки 130 и 190, благодаря чему удалось исключить строки 110 и 310. Теперь несколько видоизмененный листинг программы:

```
00010          ORG       40000
00020          ENT
00030          LD        IX,M2+2
00040          LD        DE,2
00050          XOR       A
00060          SUB       1
00070          EX        AF,AF'
00080          DI
00090          CALL      #056A
00100 M2       LD        IX,0
00120          LD        B,8
00130 M1       PUSH      BC
00140          INC       E
00150          CALL      #05A9
00160          DEC       IX
00170          DEFB      #DD          ;
00180          INC       H            ;INC   XH
00190          POP       BC
00200          DJNZ      M1
00210          DEFB      #DD          ;
00220          LD        A,H          ;LD    A,XH
00230          RRCA
00240          RRCA
00250          RRCA
00260          DEC       A
00270          AND       #03
00280          OR        #58
00290          DEFB      #DD          ;
00300          LD        H,A          ;LD    XH,A
00320          INC       E
00330          CALL      #05A9
00340          LD        IX,M2+2
00350          LD        E,2
00360          CALL      #05A9
00370          DEC       E
00380          INC       E
00390          JR        Z,M2
00400          LD        A,(M2+2)
00410          CP        1
00420          JP        #053F
Start of object:  #9C40 (40000)
Length:           #0046 (00070)
Executes:         #9C40 (40000)
```

В строках 180, 220 и 300 речь идет о старшей байте регистра I**X** (**H**igh).

Теперь некоторые комментарии. Раскладку кодового блока Вы уже знаете из предыдущей статьи. Итак, начинается загрузка. Вначале в регистр IX задается адрес того места в самой программе, где находится число, загружаемое в дальнейшем в регистр IX (строка 100 - пока что там ноль), а длина загружаемого блока в DE - два байта, то есть в строку 100 загрузится адрес первой линии загружаемого знакоместа экрана, далее (в строке 120 организуется цикл из 8 повторений для загрузки 8 пиксельных линий (байтов) знакоместа. Команда в строке 140 каждый раз обеспечивает единицу в регистре DE - число загружаемых процедурой #05A9 байтов в строке 150. Так как каждый следующий байт знакоместа имеет адрес на 256 больший, чем предыдущий, то здесь то и удобно оказалось увеличивать старший байт регистра IX, определяющего адрес загрузки для процедуры #05A9 (строки 170-180), вернув предварительно исходное (до загрузки байта процедурой #05A9) значение регистра IX в строке 160. После загрузки

всех 8 байтов знакоместа в строках 210-300 рассчитывается адрес атрибутов этого знакоместа (результат - тоже в IX), а строка 320 обеспечивает единицу в регистре DE для загрузки 1 байта атрибутов строкой 330.

В строке 340 начинается загрузка следующего знакоместа. Как и в начале программы, в IX опять заносится пара ячеек программы, куда будет загружено 2 байта, определяющие следующее знакоместо, затем в регистр E записывается двойка, но так как в D - ноль, то в DE получается 2. Затем все повторяется со следующей группой из 11 байтов.

В конце программа переходит на процедуру завершения загрузки SA/LD-RET (строка 420), как это сделала бы обычная LD-BYTES (#0556).

В заключение - дамп второго (более короткого) варианта процедуры загрузки.

9C40	DD	21	51	9C	11	02	00	AF	89
9C48	D6	01	08	F3	CD	6A	05	DD	CF
9C50	21	00	00	06	08	C5	1C	CD	C9
9C58	A9	05	DD	2B	DD	24	C1	10	7C
9C60	F4	DD	7C	0F	0F	0F	3D	E6	99
9C68	03	F6	58	DD	67	1C	CD	A9	2B
9C70	05	DD	21	51	9C	1E	02	CD	E9
9C78	A9	05	1D	1C	28	D1	3A	51	7F
9C80	9C	FE	01	C3	3F	05	00	00	BE

Дополнение ИНФОРКОМА.

Мы думаем, не будет преувеличением сказать, что этот прекрасный способ загрузки, является наиболее эффектным из всех типов нестандартных загрузчиков. Ко всему прочему здесь добавляется творческий момент на этапе кодирования блока-экрана - каждый автор может сделать эту часть работы по-своему для одного и того же экрана, и эффект при этом будет совершенно различным. Жалко только, что в связи с появлением дисководов, этот способ загрузки вынужден отправляться на пенсию. Не в силах смириться с этим, мы, как смогли, попытались исправить это положение.

Идея такая. При адаптации программы под диск надо сохранить закодированный блок-экран в том виде, в каком он был для магнитофонного варианта программы. А загружать его будет псевдо-загрузчик ("Pseudo-Loader"), имитирующий загрузку с магнитофона, сохраняя последовательность вывода знакомест на экран (только, естественно, с увеличенной скоростью). Сейчас мы предлагаем такой псевдо-загрузчик.

```
00001 ;       (C) ALANSOFT
00002 ;       "Pseudo-Loader"
00010         ORG       40000
00020         ENT
00030 WAIT    EQU       #0600          ;Задержка.
00040 BEG     CALL      #007C          ;Эта часть служит для
00050         DEC       SP             ;привязки к адресу
00060         DEC       SP             ;загрузки блока кодов.
00070         POP       IX             ;В результате в IX
00080         LD        BC,END-BEG-1   ;получим адрес,
00090         ADD       IX,BC          ;следующий за END.
00100         LD        DE,0           ;Здесь будет задана длина.
00110 LOOP    LD        L,(IX)         ;Начало основного цикла:
00120         INC       IX             ;
00130         LD        H,(IX)         ;в IX - текущий адрес
00140         INC       IX             ;в закодированном
00150         LD        B,8            ;блоке-экране;
00160 L1      LD        A,(IX)         ;
00170         LD        (HL),A         ;В HL - адрес в
00180         INC       H              ;дисплейном файле.
00190         INC       IX
00200         DJNZ      L1             ;
00210         LD        A,H            ;Расчет
00220         SUB       8              ;адреса
00230         RRA                      ;в
00240         RRA                      ;файле
00250         RRA                      ;атрибутов;
00260         AND       #03            ;результат
00270         OR        #58            ;расчета
00280         LD        H,A            ;в регистре HL.
00290         LD        A,(IX)         ;
00300         LD        (HL),A         ;
00310         INC       IX             ;
00320         LD        BC,WAIT        ;Пауза
```

```
00330 L2      DEC       BC
00340         LD        A,B           ;Проверка
00350         OR        C             ;содержимого BC
00360         JR        NZ,L2         ;на достижение нуля.
00370         LD        B,11          ;Переходим к следующему знакоместу:
00380 L3      DEC       DE            ;уменьшение счетчика на 11.
00390         LD        A,D           ;Проверка содержимого BC
00400         OR        E             ;на достижений нуля;
00410         RET       Z             ;выход из программы,
00420         DJNZ      L3            ;если ноль достигнут.
00430 END     JR        LOOP          ;Иначе – повтор основного цикла.
Start of object:  #9C40 (40000)
Length:           #0047 (00071)
Executes:         #9C40 (40000)
```

Дамп блока кодов:

9C40	CD	7C	00	3B	3B	DD	E1	01	5A
9C48	44	00	DD	09	11	00	00	DD	FC
9C50	6E	00	DD	23	DD	66	00	DD	7A
9C58	23	06	08	DD	7E	00	77	24	1B
9C60	DD	23	10	F7	7C	D6	08	1F	7C
9C68	1F	1F	E6	03	F6	58	67	DD	BD
9C70	7E	00	77	DD	23	01	00	06	08
9C78	0B	78	B1	20	FB	06	0B	1B	8F
9C80	7A	B3	C8	10	FA	18	C8	00	FB

Теперь о том, как же пользоваться псевдо-загрузчиком.

Во-первых, приведенный блок кодов "Pseudo-Loader" является релоцируемым, хотя для примера стартовый адрес равен 40000. После ассемблирования Вы можете загружать его в любое место и запускать с адреса загрузки.

Во вторых, запускаться он дол жен только будучи "присоединенным" к закодированному блоку-экрану, который располагается сразу же за 71 байтом самого псевдо-загрузчика.

В-третьих, псевдо-загрузчик (как и его прообраз) универсален: никакой предварительной настройки для конкретного закодированного блока-экрана не требуется. Длина последнего заносится В блок кодов автоматически при присоединении к псевдо-загрузчику закодированного блока-экрана.

Теперь о том, как это присоединение выполняется. Как известно, закодированный блок-экран на магнитной ленте не имеет заголовка. Перенесите его на дискету любым копировщиком типа ЛЕНТА-ДИСК. При этом, например, копировщик "AMCOPIER" присвоит блоку имя "*00"CODE. Длина его, исходя из формата данных, должна быть крат на 11 - убедитесь в этом. Теперь соедините псевдо-загрузчик, имеющий длину 71 байт с закодированным блоком "*00"CODE при помощи следующей Бейсик-программы:

```
10 BORDER 0: PAPER 0: INK 7: CLEAR 39999
20 RANDOMIZE USR 15619: REM : LOAD "pseudo" CODE 40000
30 RANDOMIZE USR 15619: REM : LOAD "*00" CODE 40071
40 LET l=PEEK 23771+256*PEEK 23772: PRINT l: POKE 40013,PEEK 23771: POKE 40014,PEEK
   23772
50 RANDOMIZE USR 15619: REM : SAVE "Filename" CODE 40000,(71+l)
60 RANDOMIZE USR 40000
```

В строке 20 загружается сам 71-байтовый псевдо-загрузчик, в строке 30 - закодированный блок, перенесенный с ленты, в строке 40 определяется длина закодированного блока, индицируется на экране и заносится в блок кодов псевдо-загрузчика. В строке 50 записывается готовый "саморазворачивающийся" релоцируемый модуль, его демонстрация в работе происходит в строке 60. Для другого блока-экрана поменяйте только имена файлов в Бейсик-строках 30 и 50.

Единственным параметром, который, возможно, потребуется подобрать - является задержка, определяющая скорость вывода картинки на экран. Числовое значение задержки задано двумя байтами по адресу 40054. Точнее, младший байт равен нулю, а подбирать достаточно только старший байт по адресу 40055. При этом скорости вывода может быть изменена в широчайших пределах. При значении, заданном в строке 30 листинга на ассемблере, обеспечивается вывод на экран заполненной на 80-90% картинки примерно за 7 секунд что, хотя и несколько замедляя процесс загрузки программы с диска, однако эффектно воспроизводит последовательную прорисовку картинки, как это было в магнитофонном варианте.

Мы продолжаем публиковать материалы, связанные с нестандартной загрузкой. На этот раз письма пришло от Владимира Кашникова.

Линейная загрузка экрана.

© *Владимир Кашников, г. Кинель, 1994.*

Загрузчик, листинг которого приводится ниже, загружает картинку плавно, линию за линией, правда длину

картинки пришлось увеличить. Конечно, это не самый: лучший из подобных загрузчиков, но зато самый простой. (*Листинги приводятся с непринципиальными изменениями ИНФОРКОМА*).

```
00010          ORG       40000
00020          ENT
00030          LD        IX,0
00040          LD        DE,17
00050          LD        A,0
00060          SCF
00070          CALL      #0556
00080          LD        IX,0
00090          LD        DE,1
00100          LD        A,255
00110          SCF
00120          DI
00130          INC       D
00140          EX        AF,AF'
00150          DEC       D
00160          CALL      #056A
00170          LD        HL,16384
00180          LD        B,3
00190 LK       PUSH      BC
00200          PUSH      HL
00210          LD        B,8
00220 KL       PUSH      BC
00230          PUSH      HL
00240          PUSH      HL
00250          POP       IX
00260          LD        B,8
00270 ST       PUSH      BC
00280          LD        DE,32
00290          PUSH      HL
00300          PUSH      IX
00310          PUSH      DE
00320          LD        IX,0
00330          LD        DE,1
00340          CALL      #05A9
00350          POP       DE
00360          POP       IX
00370          CALL      #05A9
00380          POP       HL
00390          INC       H
00400          PUSH      HL
00410          POP       IX
00420          POP       BC
00430          DJNZ      ST
00440          POP       HL
00450          LD        DE,32
00460          ADD       HL,DE
00470          POP       BC
00480          DJNZ      KL
00490          POP       HL
00500          LD        DE,2048
00510          ADD       HL,DE
00520          POP       BC
00530          DJNZ      LK
00540          LD        IX,0
00550          LD        DE,1
00560          CALL      #05A9
00570          LD        IX,22528
00580          LD        DE,768
00590          CALL      #05A9
00600          EI
00610          RET
Start of object:  #9C40 (40000)
```

```
Length:           #0074 (00116)
Executes:         #9C40 (40000)
```

Далее - программа, конвертирующая стандартный экран в вид, пригодный для данного загрузчика:

```
10 LOAD "filename" CODE 16384
20 LET l=16384
30 LET k=41000
40 POKE k,0: LET k=k+1
50 FOR h=1 TO 3
60 FOR f=1 TO 8
70 FOR n=1 TO 8
80 POKE k,0: LET k=k+1
90 FOR m=0 TO 31
100 POKE k,PEEK (l+m)
110 LET k=k+1
120 NEXT m
130 LET l=l+256
140 NEXT n
150 LET l=l-2016
160 NEXT f
170 IF h=1 THEN LET l=18432
180 IF h=2 THEN LET l=20480
190 NEXT h
200 POKE k,0: LET k=k+1
210 FOR n=0 TO 767
220 POKE k,PEEK (22528+n)
230 LET k=k+1
240 NEXT n
250 BEEP 1,0
260 SAVE "name" CODE 41000,7106
```

Комментарий ИНФОРКОМА.

Загрузчик, действительно, почти самый простой, из всех, которые нам приходилось встречать. Реализовать его удалось благодаря тому, что в основе лежит использование процедуры ПЗУ #05A9 LD-LOOP - это вершина цикла загрузки последовательности байтов. На входе в эту процедуру должны быть заданы: в IX - адрес загрузки; в DE - число загружаемых байтов. Однако, процедура используется уже тогда, когда зафиксирован и обработан пилоттон и установлена синхронизация с сигналом, поступающим с магнитофона. То есть, когда, как говорится, процесс уже пошел. Кроме того, начальный (флаговый) байт поступающей кодовой последовательности 0 или 255 - определяет тип загружаемого блока - заголовок или блок кодов. Загрузка происходит "пакетами" по 32 байта - это максимальная непрерывная последовательность, учитывающая раскладку дисплейного файла, то есть одна пиксельная линия экрана. Причем для имитации флагового байта, каждый пакет должен состоять из 33 байтов, независимо от значения этого самого фиктивного флагового байта.

Рассмотрим, как же происходит загрузка при помощи этого загрузчика. Отметим сразу, что возможно использование экранного блока кодов, как с заголовком, так и без него. Строки 30-70 (см. листинг на Ассемблере) выполняют фиктивную загрузку 17 байтов заголовка стандартной процедурой LD-BYTES. Фиктивную потому, что адрес загрузки равен нулю, то есть загрузка идет в область ПЗУ. Можно изменить строку:

```
00050           LD        A,255
```

и загрузка заголовка вообще не произойдет. А если будете использовать блок кодов без заголовка, то можете взять да и удалить строки 30-70, еще более укоротив, загрузчик.

Далее. Строки 80-110 - в них идет подготовка исходных данных для загрузки блока кодов длиной 1 байт в адрес 0. Эта часть, судя по адресу в IX, как Вы понимаете, тоже является фиктивной загрузкой и нужна для синхронизации с пилоттоном и успешного начала того самого процесса. Но в этом фрагменте происходит одна нефиктивная вещь - загрузка настоящего флагового байта (255), определяющего, что загружаемый блок является не заголовком, а блоком кодов.

Собственно загрузка заданного в DE одного байта происходит при помощи вызова части процедуры LD-BYTES (строка 160) и предшествующих ей нескольких команд в строках 120-150, копирующих начальный фрагмент LD-BYTES (по адресам #0556-#0559). Суть такой замены вот в чем: исключается та часть процедуры LD-BYTES, где задается на стеке адрес завершения процедуры (#053F) для перехода на SA/LD-RET.

Итак, зафиксирован пилоттон, принят настоящий флаговый байт, он равен 255, то есть загружается блок кодов, далее начинается загрузка и, согласно условию, загружается один байт по адресу 0. При этом программа, после возврата по RET, не завершает загрузку, как это сделала бы LD-BYTES переходом на SA/LD-RET, а попадает на строку 170. А между тем, поток информации с магнитофона продолжается.

Только теперь начинается загрузка дисплейного файла. В HL начальный адрес загрузки, далее - три вложенных цикла: загрузка трех сегментов экрана по 8 строк по 8 пиксельных линий - каждая. Загрузка, как уже говорилось, происходит пакетами по 32, а точнее, по 33 байта (вместе с фиктивным флаговым байтом). Такая

раскладка кодового блока предусматривается при кодировании стандартного экрана. Считывание флагового байта пакета происходит в строках 320-340 (причем, опять же, фиктивное, но оно необходимо, так как чтение флагового байта предусмотрено процедурой с точкой входа #05A9 - LD-LOOP). А считывание информационных 32 байт про исходит в строках 350-370, после чего следует переход к следующей пиксельной линии1 (адреса в HL увеличивается на 256 байтов в строке 390).

После загрузки всех 6144 байтов дисплейного файла надо загрузить файл атрибутов. Здесь загрузка выполняется аналогичным образом, только используется один пакет, так как все 768 байтов файла атрибутов идут подряд. Сначала (в строках 540-560) идет фиктивная загрузка флагового байта, (уже ясно, для чего), далее загрузка файла атрибутов (строки 570-590).

Посмотрим теперь, какой же расклад должен иметь закодированный кодовый блок-экран. 3 сегмента по 8 строк по 8 линий по 32 байта, то есть 3*8*8=192 пакета по 32 байта (в соответствующей последовательности) с предшествующими им фиктивными флаговыми байтами увеличат длину закодированного блока на 192 байта, плюс один фиктивный (нефлаговый) байт в начале блока и плюс еще один фиктивный флаговый байт, пред шествующий файлу атрибутов. Итого получается: 6912 + 192 + 2 = 7106 байтов. Такая кодировка и выполняется приведенной выше Бейсик-программой, а фиктивные байты вставляются Бейсик-строками 40, 80 и 200. причем, поскольку байты эти - фиктивные, совершенно не важно, что вы занесете в ячейки: можете задать: POKE к,255 - это абсолютно ни на что не повлияет, так как значение флагового байта в память не заносится. Однако Вы можете "увидеть" процесс загрузки фиктивных байтов, задав в регистре IX, например, адрес последней ячейки атрибутов. Для этого измените строку 320:

```
00320         LD        IX,23295
```

при этом получите эффект хаотичного мигания правого нижнего знакоместа экрана при загрузке картинки.

Следует сказать, что при записи кодового массива командой SAVE будет добавлен еще и настоящий флаговый байт, равный 255, определяющий, что это блок кодов, а не заголовок.

Заметим, что, несмотря на длину загружаемого файла, большую на 194 байта, чем обычный экран, сам загрузчик короче, чем аналогичные, поэтому несколько большее время при загрузке экрана компенсируется меньшим временем при загрузке самого загрузчика.

Теперь, по традиции, приводим шестнадцатеричный дамп загрузчика с контрольными суммами:

```
9C40 | DD  21  00  00  11  11  00  3E | 3A
9C48 | FF  37  CD  56  05  DD  21  00 | 40
9C50 | 00  11  01  00  3E  FF  37  F3 | 65
9C58 | 14  08  15  CD  6A  05  21  00 | 82
9C60 | 40  06  03  C5  E5  06  08  C5 | C2
9C68 | E5  E5  DD  E1  06  08  C5  11 | 70
9C70 | 20  00  E5  DD  E5  D5  DD  21 | A6
9C78 | 00  00  11  01  00  CD  A9  05 | A1
9C80 | D1  DD  E1  CD  A9  05  E1  24 | 2B
9C88 | E5  DD  E1  C1  10  E0  E1  11 | 6A
9C90 | 20  00  19  C1  10  D1  E1  11 | F9
9C98 | 00  08  19  C1  10  C5  DD  21 | E9
9CA0 | 00  00  11  01  00  CD  A9  05 | C9
9CA8 | DD  21  00  58  11  00  03  CD | 7B
9CB0 | A9  05  FB  C9  00  00  00  00 | BE
```

В том случае, если Вы будете использовать картинку, как блок кодов без заголовка, то набор и запуск загрузчика производите с адреса #9C4D (40013). Заметим также, что загрузчик этот - **релоцируемый**, возможна загрузка его под любой адрес и запуск с адреса загрузки.

В заключение, мы хотим обратить Ваше внимание на то, что существуют специальные приемы для того, чтобы избежать загрузки большого числа фиктивных флаговых байтов. Предлагаем на досуге, в качестве "зарядки для мозгов", подумать об этом. Можем только заверить Вас, что решение существует, и довольно простое, хотя и не совсем очевидное. В качестве информации к размышлению можете использовать предыдущий материал Павла Бобровникова - там тоже используется процедура с точкой входа #05A9, однако без загрузки фиктивных байтов. Если Вы "прочувствуете" принцип построения этих загрузчиков и найдете ключ к использованию процедуры #05A9, то сможете без проблем создавать загрузчики нового поколения минимальной длины. Это своего рода RAXXLA - "дорога к другим галактикам" - где Вы получаете полную свободу для загрузки чего угодно, куда угодно и в каком угодно порядке.

Можно также попробовать непосредственное использование процедуры загрузки 8 битов и формирования из них байта: LD-8-BITS (#05CA-#05E2). Можете предусмотреть также контроль правильности считывания. В общем - пишите! А лучший вариант (или варианты) мы обязательно опубликуем в ZX-РЕВЮ в новом 1995 году.

Расширение памяти "Ленинграда-1" до 128К

© Беззубцев А.С., г. Зеленогорск (Красноярск-45), 1994.

Под таким названием в ZX-РЕ-ВЮ-94, № 4 (стр.26) была опубликована моя статья. В ней я совершенно упустил из виду вопрос о подключении ПЗУ. Кроме того, была обнаружена ошибка в схеме подключения музыкального процессора (далее - сопроцессор).

Сначала о подключении ПЗУ.

Чтобы использовать микросхему 27256 (РФ8) с прошивкой SOS48 и SOS128, необходимо на плате компьютера сделать следующие изменения:

27 вывод ROM1 (D30) отрезать от A13 Z80 и адресного диода клавиатуры. Затем диод соединить с A13 Z80, минуя 27 вывод. После этого на 27 вывод подают сигнал ROM с 01(12) (см. схему в РЕВЮ № 4).

Элемент D7.2 (см. схему в РЕВЮ № 4) освободить и включить согласно рисунку 1. Сигнал RDROM, если он был разорван, восстановить.

ПЗУ с TR-DOS из контроллера дисководов необходимо удалить.

Если в контроллере дисководов на шине данных стоит буфер типа АП6, то 19 вывод этой микросхемы освободить и на него подать сигнал IORQ с Z80.

Теперь о сопроцессоре. Было замечено, что некоторые программы, нормально работающие на переработанном "Ленинграде-1" без сопроцессора, отказывались работать на компьютерах с ним. Причиной столь странного поведения оказалась некорректная схема подключения сопроцессора, в которой не учитывался цикл подтверждения прерывания Z80. Сопроцессор во время этого цикла мог "засорять" шину данных. Это происходило только если прерывание произошло на некоторых адресах из #C000-#FFFF.

Надо отметить, что не один я не учел цикл подтверждения прерывания. Мной были проверены несколько схем подключения сопроцессора, в том числе и из "платы расширения до 128К", которая распространялась ИНФОРКОМом, схема Анатолия Гармаисанова из книги ZX-ФОРУМ. Из всех проверенных схем только одна не "засоряла" шину данных - это схема подключения сопроцессора в компьютере "ATM-Turbo". Я предполагаю, что во всех схемах подключения сопроцессора, где используется сигнал M1 Z80, сопроцессор будет "засорять" шину данных.

Решение проблемы довольно простое: необходимо в разрыв сигнала IORQ с Z80 добавить два элемента, включив их согласно рисунку 2 - эта схема предназначена для доработки моей схемы подключения сопроцессора, поэтому в ней оправдано применение двух разных логических элементов. В других случаях это может оказаться неудобным. Поэтому, для доработки других схем лучше использовать одну микросхему ЛЕ1 (рисунок 4).

Еще несколько уточнений.

Элементы, обведенные штриховой линией, это свободные элементы микросхемы, ЛН1 на плате компьютера.

Если Вам не нужен сопроцессор, то микросхему D9 - не ставьте.

К сожалению, в своей публикации "Расширение памяти "Ленинграда- 1" до 128К" я не указал что:

- сигнал RDROM необходимо разорвать (только если Вы используете 27256);
- 6, 10 и 13 выводы микросхемы D36 (КПП) соединить с общим проводом;
- нагрузочные резисторы на KEMPSTON-джойстик должны иметь сопротивление 1-2 кОм и соединяться не с +5В, а с общим проводом;
- общий провод джойстика надо также соединить с +5В.

Если Вам надо отключать ПЗУ, и Вы поставили 27512. то в разрыв сигнала RDROM включите схему, приведенную на рисунке 3. Сигнал ROMCS отключает ПЗУ компьютера, когда он в "1".

Если у Вас возникнут вопросы или замечания по моей схеме расширения памяти, то жду писем по адресу:

663690, Красноярский край, г. Зеленогорск (Красноярск-45),
ул.Бортникова, д.36, кв.52. Беззубцеву Алексею Сергеевичу.

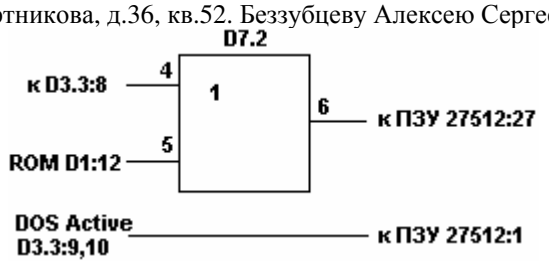


Рис. 1

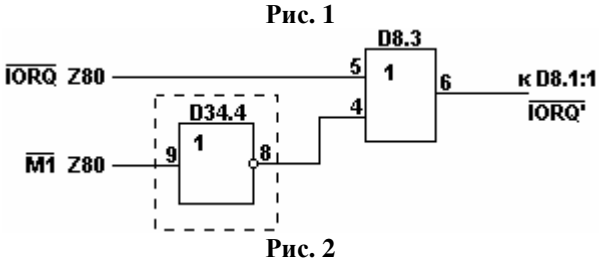


Рис. 2

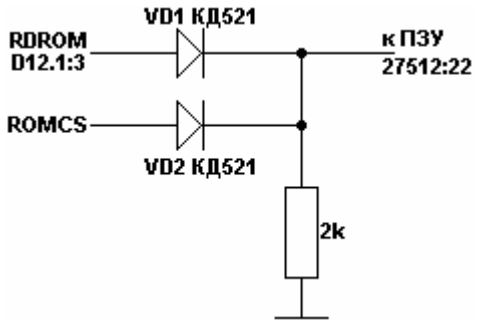


Рис. 3

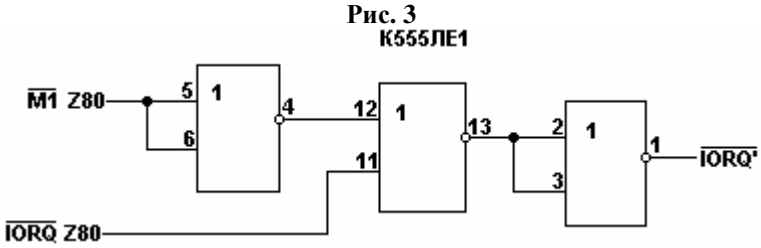


Рис. 4

Некоторые подробности о мониторе TURBO-90

© Игорь Назаров (IWN), г. Нижний Новгород, 1994.

Вначале об аналогах дополни тельных команд монитора TURBO-90, которыми интересовался, в частности, Вадим Бодров.

1. Эквивалент команды "," по лучить не удалось, поскольку при выполнении функции USR addr в регистр A процессора заносится младший байт адреса, а не ноль.

Поэтому придется использовать примитивную программу, которую можно ввести хотя бы в мониторе TURBO-90 (далее - MT90):

```
addr     AF          XOR   A
         C3 1E 39    JP    14622
```

Вызов программы:

```
RANDOMIZE USR addr
```

При этом результат выводится в любую часть экрана, то есть:

ВНИМАНИЕ! В РЕВЮ-94, № 4, стр. 27, в схеме на рис.1 не указаны входы элемента D7.3 - должны быть 9 и 10. Кроме того, неправильно указан выход элемента D8.3 - дол жен быть 8, а не 11, как указано на схеме.

```
PRINT: RANDOMIZE USR addr – в верхнюю;
PRINT #0: RANDOMIZE USR addr – в нижнюю;
LPRINT: RANDOMIZE USR addr – на принтер.
```

Только вот выдает она результат на 18-25 байтов меньше, чем просто ",". Это связано с длиной строки вызова. Истинное значение свободной памяти будет равно по лученному результату плюс длина информации в командной строке.

2. Эквивалент команды "*". Вход в MT90:

а). RANDOMIZE USR 132

б). RANDOMIZE USR 96

в). RANDOMIZE USR 14648

3. Эквивалент команды "-". Установка турборежима.

а). POKE 23604,82: POKE 23605,75

б). RANDOMIZE USR 14639

4. Эквивалент команды "+". Отмена турборежима.

а). Достаточно одного POKE 23605,N (N<>75)

б). RANDOMIZE USR 14615

5. Эквивалент команды "("Вывод параметров заголовков файлов при загрузке с ленты.

а). RANDOMIZE USR 14596

б). RANDOMIZE USR 14593

в). POKE 23658,32 - параметры будут выводиться в десятичном виде;

г). POKE 23658,160 - параметры будут выводиться в 16-ричном виде.

6. Эквивалент команды ")". Отмена вывода параметров заголовков файлов.

а). RANDOMIZE USR 14604

б). POKE 23658,0

Результат действия команд "(", ")" (и ещё один эффект) достигаются с помощью системной переменной FLAGS2 (23658):

Бит 7=1 - 16-ричный вид чисел в MT90 При командах "," и "("

Бит 7=0 - десятичный вид чисел.

Бит 6=1 - выключение цифрового дампа памяти, расширенное поле ASCII в MT90.

Бит 6=0- вывод в MT90 дампа памяти в DEC/HEX виде (в соответствии с битом 7).

Бит 5=1- вывод параметров заголовка файла в команде "(".

Бит 5=0 - отключение печати параметров файла - команда ")".

Биты 4…0 - используются Бейсиком. При установке параметров их можно игнорировать.

Режим вывода DEC/HEX можно переключать и так:

RANDOMIZE USR 14982

это инвертирует бит 7 переменной FLAGS2, а режим DUMP+ASCII или только ASCII - с помощью:

RANDOMIZE USR 14995

это инвертирует бит 6 переменной FLAGS2.

7. Полный сброс с очисткой памяти происходит при:

а). RANDOMIZE USR 99

б). Нажатие RESET и CAPS SHIFT

в). Нажатие на RESET, если байт по адресу 23568 не равен 1 (подробнее об этом - ниже).

В дополнение к статье в ZX-РЕВЮ-94, № 3, стр. 21-22, сообщаю:

1. MT90 хранит адрес начала просматриваемой памяти по адресу #5CAA (23722), а это, как всем известно, находится в памяти калькулятора.

MT90 также портит при вызове слово по адресу #5CAC (23724), используя его для хранения указателя на текущий регистр, а также слово по адресу #5CAE (23726) -для хранения указателя стека SP.

2. Команда MT90 "(" (SS+8) - возврат в Бейсик без очистки экрана, но с прекращением выполнения программы, то есть операторы, идущие за "*", не выполняются. Это достигается переходом на адрес #1303 из монитора.

3. Команда MT90 ")" (SS+9) полностью равнозначна:

```
RST     8
DEFB    #FD
```

Это выдача 253-го сообщения из таблицы по адресу #1391, которая содержит всего-то 28 сообщений об ошибках и еще одно, которое выдаётся сразу же после сброса. Неужели создателям не пришло в голову число поменьше?

4. MT90 при запуске вызывает подпрограмму #1255 (04699), вызывающуюся только при сбросе (в том числе и без очистки памяти), которая :

а). восстанавливает исходное значение системной переменной CHARS - адреса, определяющего символьный набор;

б). включает режим прерываний IM 1;

в). восстанавливает значение регистра IY;

г). восстанавливает системные переменные REPDEL и REPPER;

д). восстанавливает системную переменную CHANS;

е). копирует из ПЗУ область адресов, подключенных к потокам каналов STRMS, а также область информации о каналах, при этом, записывая маркер её конца (#80) не туда, куда надо, изменяя информацию в первом дополнительно созданной блоке информации о канале.

Поэтому, если Вы подключаете новые каналы или имеете программное обеспечение, которое это делает, **будьте осторожны** при использовании MT90 и кнопки RESET - при сбросе без очистки памяти также почти все восстанавливается. Однако для предотвращения неприятностей можете создать "пустышку" - неиспользуемый канал.

Кстати, Вам писали, что в ПЗУ TURBO-90 исправлена ошибка обработки немаскируемого прерывания NMI, так посмотрите - и увидите, что это больше походит на рудименты и атавизмы. При нажатии на кнопку "NMI" Вы вывалитесь в MT90, в большинстве случаев бесполезный или зависающий.

Поток STRMS #FD, что по адресу #5C10 (23568) лучше не переключать, поскольку, если по адресу #5C10 содержится не 1, то, как только Вы нажмете кнопку RESET (даже без CAPS SHIFT), то обнаружите память своего Spectrum'а изумительно чистой, как асфальт после ливня.

Отсюда выводы:

ПЕРВЫЙ, и самый главный: монитор TURBO-90 в большинстве реальных случаев бесполезен, но лучше иметь его, чем ничего.

ВТОРОЙ: чтобы образумить любителей щелкать кнопкой RESET, считающих, что в TURBO-90 всё можно, следует переустанавливать STRMS #FD, но не с помощью

POKE 23568,250: POKE 23569,184

которые без ножа убивают Бейсик, а с помощью простенькой программки Rock'n'Roll:

```
1 REM FREEWARE
2 LET a=PEEK 23635+256*PEEK 23636+5
3 FOR b=a TO a+6
4 READ c
5 POKE b,c
6 NEXT b
7 PRINT "Press RESET for Rock'n'Roll"
```

```
8 DATA 33,250,184,34,16,92,201
9 RANDOMIZE USR a
```

Вот её дизассемблированный вид:

```
addr  21 FA B8   LD    HL,#B8FA
      22 10 5C   LD    (#5C10),HL
      C9         RET
```

А если совместно с ней использовать защиту от остановки программы? Например, отключение BREAK или установка количества строк в нижней части экрана равным нулю. Прощай тогда, основное достоинство прошивки ПЗУ TURBO-90…

Полезные подпрограммы MT90, которых очень мало:

#392F - установка турбо-режима.

#3BF2 - проверка на турбо-режим. Если он установлен, то будет выставлен флаг Z регистра F.

#39C7 - печать числа, находящегося в DE, в шестнадцатеричном виде. Если при входе регистр A не равен нулю, то выводится только один байт, находящийся в регистре E.

#38C1 - вывод значения регистра A в 16-ричнвм виде.

#3B03 - очистка экрана, восстановление всех регистров со стека и передача управления по RET.

#3B33 - INKEY (использует системную переменную LAST-KEY). Если прерывания запрещены, то эта подпрограмма успешно "подвесит" Ваш компьютер.

#3904 - включение режима печати параметров заголовка файла. Оператор "(".

* * *

Этюды

Все программы, приведённые в этом разделе, были набраны при помощи ассемблера "ZEUS +", версии V 3.1 ассемблера ZEUS, которую доработал Сергей Ханцис из Ижевска (поставляется в составе дистрибутивной дискеты "Утилиты-2"). Можем сказать, что это самая лучшая полностью дисковая версия из тех, которые нам приходилось встречать.

Стартовый адрес для всех блоков для примера выбран 40000. Естественно, Вы можете задать своё значение.

Для устранения возможных неоднозначностей, для каждой программы приводится шестнадцатеричный дамп с контрольными суммами.

© *Патрахин В.В., г. Оленегорск, 1994.*

Предлагаю в Вашу новую и замечательную рубрику ЭТЮДЫ несколько своих программ. Первые три небольшие процедурки я написал под впечатлением программ для расчёта адреса знакоместа в экранном файле А.А. Иванова (автора и основоположника этой рубрики, очень понравившейся нашим читателям, судя по письмам - прим. ИФК). А мои программы рассчитывают адрес знакоместа в файле атрибутов.

Первый вариант. На входе: в E - номер строки, в D - номер столбца. На выходе: DE - без изменений; в HL - адрес в файле атрибутов.

```
00010             LD        A,E
00020             AND       #18
00030             RRCA
00040             RRCA
00050             RRCA
00060             OR        #58
00070             LD        H,A
00080             LD        A,E
00090             AND       #07
00100             RRCA
00110             RRCA
00120             RRCA
00130             ADD       A,B
00140             LD        L,A
```

Второй вариант. На входе: в L - строка, в H - столбец. На выходе: в HL - адрес в файле атрибутов.

```
00010             LD        A,L
00020             AND       #07
00030             RRCA
00040             RRCA
00050             RRCA
00060             ADD       A,H
00070             LD        H,L
00080             LD        L,A
00090             LD        A,H
00100             AND       #18
00110             RRCA
00120             RRCA
00130             RRCA
00140             OR        #58
00150             LD        H,A
```

Хотя процедуры, приведенные В.В. Патрахиным, чуть длиннее нашей (см. "Элементарная графика", стр. 86), зато придуманы самостоятельно, без подсказок. А это наиболее ценно. Однако, Вернёмся к письму.

А эта программа переводит адрес в экранном файле в адрес в файле атрибутов.

На входе: в HL - адрес в экранном файле. На выходе - в HL -адрес в файле атрибутов.

```
00010             LD        A,H
00020             RRCA
00030             RRCA
00040             RRCA
00050             AND       #03
00060             OR        #58
00070             LD        H,A
```

Следующая программа предназначена для очистки заданного окна экрана так, как это происходит в игре ROBOCOP. В программе используются следующие переменные:

YX - двухбайтовая величина, старший байт - строка, младший - столбец левого верхнего угла очищаемого окна.

VERT - размер по вертикали (в знакоместах) окна.
HORIZ - то же по горизонтали.

```
00010              ORG       40000
00020              ENT
00030 YX           EQU       #0404
00040 VERT         EQU       16
00050 HORIZ        EQU       24
00060 PAUSE        EQU       50
00070              LD        HL,YX
00080              LD        A,L
00090              AND       #07
00100              RRCA
00110              RRCA
00120              RRCA
00130              ADD       A,H
00140              LD        H,L
00150              LD        L,A
00160              AND       #18
00170              OR        #40
00180              LD        H,A
00190              LD        B,2
00200 L4           PUSH      BC
00210              PUSH      HL
00220              LD        C,VERT
00230 L3           LD        B,4
00240              PUSH      HL
00250              LD        DE,#20
00260              ADD       HL,DE
00270              EX        DE,HL
00280              POP       HL
00290              LD        A,D
00300              CP        H
00310              JR        Z,L1
00320              PUSH      HL
00330              LD        DE,#0720
00340              ADD       HL,DE
00350              EX        DE,HL
00360              POP       HL
00370 L1           PUSH      BC
00380              PUSH      HL
00390              LD        B,HORIZ
00400              XOR       A
00410 L2           LD        (HL),A
00420              INC       HL
00430              PUSH      BC
00440              LD        B,PAUSE
00450 PAUS         DJNZ      PAUS
00460              POP       BC
00470              DJNZ      L2
00480              POP       HL
00490              POP       BC
00500              INC       H
00510              INC       H
00520              DJNZ      L1
00530              EX        DE,HL
00540              DEC       C
00550              JR        NZ,L3
00560              POP       HL
00570              POP       BC
00580              INC       H
00590              DJNZ      L4
00600              RET
Start of object    : #9C40 (40000)
Length             : #004A (00074)
```

```
Executes          : #9C40 (40000)
```

И дамп блока кодов:

```
9C40 | 21  04  04  7D  E6  07  0F  0F | 8D
9C48 | 0F  84  65  6F  E6  18  F6  40 | 7F
9C50 | 67  06  02  C5  E5  0E  10  06 | 29
9C58 | 04  E5  11  20  00  19  EB  E1 | F3
9C60 | 7A  BC  28  07  E5  11  20  07 | 7E
9C68 | 19  EB  E1  C5  E5  06  18  AF | 60
9C70 | 77  23  C5  06  32  10  FE  C1 | 72
9C78 | 10  F6  E1  C1  24  24  10  EB | FF
9C80 | EB  0D  20  D3  E1  C1  24  10 | DD
9C88 | CA  C9  00  00  00  00  00  00 | B7
```

© Александр Шишкин, г. Ярославль, 1994.

Хочу поделиться с Вами моими программами и буду рад, если они кому-нибудь пригодятся.

Первая программа выполняет плавное затухание экрана.

```
00010             ORG       40000
00020             ENT
00030             LD        B,8
00040             LD        C,#C7
00050 LOOP        LD        A,C
00060             SUB       8
00070             LD        (COM),A
00080             LD        C,A
00090             LD        HL,22528
00100             LD        DE,768
00110 L1          LD        A,(HL)
00120             DEFB      #CB
00130 COM         DEFB      0
00140             LD        (HL),A
00150             INC       HL
00160             DEC       DE
00170             LD        A,D
00180             OR        E
00190             JR        NZ,L1
00200             PUSH      BC
00210             LD        BC,#1800
00220 L2          DEC       BC
00230             LD        A,B
00240             OR        C
00250             JR        NZ,L2
00260             POP       BC
00270             DJNZ      LOOP
00280             RET
Start of object   : #9C40 (40000)
Length            : #0028 (00040)
Executes          : #9C40 (40000)
```

Эта программа выключает биты в значениях атрибутов, отвечающие сначала за мерцание, затем яркость и т.д. В связи с укорочением процедуры, применен приём, при котором программа в процессе работы сама себя видоизменяет.

Посмотрите: в строках 40-50 при помощи регистра C в аккумулятор заносится код #C7. В процессе цикла, организованного в строке 30, из этого кода (#C7) каждый раз вычитается 8 (строка 60). При этом получаются числа: #BF, #B7, #AF, … #87. Таким образом, вместе с префиксом #CB в строке 120, в памяти последовательно во времени генерируются команды: RES 7,A; RES 6,A; RES 5,A; … RES 0,A, которые затем и выполняются программой.

Благодаря такому приему удалось получить длину программы всего 40 байтов, однако победителем (39 байтов) среди процедур, реализующих этот эффект, всё же пока является программа из игры TERMINATOR-2, присланная Кривцовым А.И. (ZX-РЕВЮ-94, № 4, стр. 36), к тому же она релоцируемая. Так-так, посмотрим, удастся ли нашим программистам превзойти западных…

В комплекте с первой процедурой предлагаю обратную - которая плавно "зажигает" картинку. Переменная ADRES - адрес расположения картинки в памяти.

```
00010             ORG       40040
00020             ENT
00030 ADRES       EQU       ENDR+1
00040             LD        HL,ADRES
```

```
00050             LD        DE,16384
00060             LD        BC,6144
00070             LDIR
00080             LD        IX,ENDR
00090             LD        B,8
00100 LOOP        PUSH      BC
00110             LD        A,#C9
00120             LD        (IX+0),A
00130             DEC       IX
00140             DEC       IX
00150             LD        HL,ADRES+6144
00160             LD        DE,22528
00170             LD        BC,768
00180 L1          LD        A,(HL)
00190             CALL      REST
00200             LD        (DE),A
00210             INC       DE
00220             INC       HL
00230             DEC       BC
00240             LD        A,B
00250             OR        C
00260             JR        NZ,L1
00270             LD        BC,#1800
00280 L2          DEC       BC
00290             LD        A,B
00300             OR        C
00310             JR        NZ,L2
00320             POP       BC
00330             DJNZ      LOOP
00340             LD        B,8
00350 L3          LD        A,#CB
00360             INC       IX
00370             INC       IX
00380             LD        (IX+0),A
00390             DJNZ      L3
00400             RET
00410 REST        RES       7,A
00420             RES       6,A
00430             RES       5,A
00440             RES       4,A
00450             RES       3,A
00460             RES       2,A
00470             RES       1,A
00480 ENDR        RET
Start of object   : #9C40 (40000)
Length            : #0058 (00088)
Executes          : #9C40 (40000)
```

В этой процедуре используется аналогичный прием для сокращения размера программы: здесь модифицируется часть программы в строках 410-480. В последовательности двухбайтовых команд RES N,A в подпрограмме REST вместо префикса подставляется команда RET (см. строки 80-140). Из-за этого часть подпрограммы REST отсекается. Затем (строки 340-390) исходный вариант программы восстанавливается.

После тестирования этих программ сама собой напросилась идея совместного их использования. В строке 10 второго листинга Вы видите команду ORG 40040 - длина первой процедуры 40 байтов. Так как длина второй - 88 байтов, то адрес загрузки картинки будет равен 40128. Испытание можете провести при помощи простейшей Бейсик-программы:

```
10 BORDER 0: PAPER 0: INK 0: POKE 23624,0: CLS
20 LOAD "filename" CODE
30 LOAD "screen" CODE 40128
40 PAUSE 0
50 RANDOMIZE 40040
60 PAUSE 0
70 RANDOMIZE 40000
70 GO TO 40
```

Шестнадцатеричный дамп объединённого варианта программ, файл "filename", загружаемый в 20 Бейсик-

строке:

```
9C40 | 06  08  0E  C7  79  D6  08  32 | 48
9C48 | 53  9C  4F  21  00  58  11  00 | AC
9C50 | 03  7E  CB  00  77  23  IB  7A | 67
9C58 | B3  20  F6  C5  01  00  18  0B | A6
9C60 | 78  B1  20  FB  C1  10  DD  C9 | B7
9C68 | 21  C0  9C  11  00  40  01  00 | D3
9C70 | 18  ED  B0  DD  21  BF  9C  06 | 20
9C78 | 08  C5  3E  C9  DD  77  00  DD | 19
9C80 | 2B  DD  2B  21  C0  B4  11  00 | F5
9C88 | 58  01  00  03  7E  CD  B1  9C | 18
9C90 | 12  13  23  0B  78  B1  20  F4 | BC
9C98 | 01  00  18  0B  78  B1  20  FB | 9C
9CA0 | C1  10  D6  06  08  3E  CB  DD | D7
9CA8 | 23  DD  23  DD  77  00  10  F5 | C0
9CB0 | C9  CB  BF  CB  B7  CB  AF  CB | 66
9CB8 | A7  CB  9F  CB  97  CB  8F  C9 | EA
```

Александр Шишкин просит опубликовать его адрес, что мы и делаем. Он готов помочь другим хакерам тем, что знает и хочет спросить то, чего не знает. Александр предлагает: нестандартные загрузчики, интерфейс, аналогичный "стрелке" в ARTSTUDIO (только без стрелки), с управлением от всех джойстиков одновременно, и многие другие разработки.

150062, г. Ярославль, ул. Космонавтов, д.7, кв.20
Шишкин А.П. (SHURa'94)

© Toadstool & Ter.Val". hackers, г. Ухта, (Коми), 1994.

Хотим предложить программу печати символами двойной высоты. Она короче тех, которые Вы печатали. Процедура для печати 32-х символов взята в одном из загрузчиков доработана нами для печати 384-х символов.

Особенность программы заключается в том, что текст, который нужно напечатать, располагается в памяти задом наперед и из кода символа надо вычесть число 32. В строке 10 задается адрес расположения текста в памяти (так как текст расположен задом наперед, указывать надо последний адрес). Если в тексте (с учетом вычитания 32) встретится код 222, то позиция печати будет перенесена на следующую строку. Если встретится код 223, это означает завершение печати и выход из процедуры.

Листинг приводится с непринципиальными изменениями.

```
00010 TEXT        EQU       41037
00015 PRINT       EQU       16384
00020             ORG       40000
00030             ENT
00040             LD        HL,PRINT
00050             LD        (L1+1),HL
00060             LD        HL,TEXT
00070             LD        (L2+1),HL
00080 L1          LD        DE,0
00090 L2          LD        HL,0
00100 L3          LD        A,(HL)
00110             CP        222
00120             JR        Z,L6
00130             CP        223
00140             RET       NC
00150             RLCA
00160             PUSH      HL
00170             PUSH      DE
00180             LD        L,A
00190             LD        H,15
00200             ADD       HL,HL
00210             ADD       HL,HL
00220             INC       H
00230             CALL      L4
00240             EX        DE,HL
00250             LD        BC,#F820
00260             ADD       HL,BC
00270             EX        DE,HL
00280             CALL      L4
00290             POP       DE
```

```
00300             INC       E
00310             POP       HL
00320             DEC       HL
00330             JR        L3
00340 L4          LD        B,4
00350 L5          LD        A,(HL)
00360             RRCA
00370             OR        (HL)
00380             INC       HL
00390             OR        (HL)
00400             LD        (DE),A
00410             INC       D
00420             LD        (DE),A
00430             INC       D
00440             DJNZ      L5
00450             RET
00460 L6          DEC       HL
00470             LD        (L2+1),HL
00480             LD        HL,(L1+1)
00490             LD        DE,64
00500             ADD       HL,DE
00510             LD        (L1+1),HL
00520             JR        L1
Start of object   : #9C40 (40000)
Length            : #0053 (00083)
Executes          : #9C40 (40000)
```

Дамп блока кодов:

9C40	21	00	40	22	4D	9C	21	4D	B6
9C48	A0	22	50	9C	11	00	00	21	C4
9C50	00	00	7E	FE	DE	28	2C	FE	98
9C58	DF	D0	07	E5	D5	6F	26	0F	08
9C60	29	29	24	CD	75	9C	EB	01	3C
9C68	20	F8	09	EB	CD	75	9C	D1	BF
9C70	1C	E1	2B	18	DD	06	04	7E	B1
9C78	0F	B6	23	B6	12	14	12	14	FE
9C80	10	F5	C9	2B	22	50	9C	2A	4D
9C88	4D	9C	11	40	00	19	22	4D	E6
9C90	9C	18	B9	00	00	00	00	00	99

В приведенной процедуре печати для сокращения объёма тоже приценяется приём модификации программы в процессе работы. Вообще говоря, модифицируются не команды, а данные: изменяются значения, задаваемые в регистры DE и HL в строках 90 и 100. Вместо нулей при работе подставляются конкретные данные.

Не знаем, хорошо или плохо то, что текст для печати должен быть "закодирован" - видимо, это зависит от конкретных задач. Слово "закодирован" взято в кавычки, поскольку, если текст должен быть напечатан строчными буквами, то при "кодировании" они просто должны быть заменены на заглавные. Так или иначе, упрощение процедуры привело к усложнению процесса подготовки текста для печати. Но, как говорится, о вкусах не спорят. От себя мы можем дополнить процедуру программой на Бейсике, кодирующей текстовые сообщения.

```
10 LET begin=41000: LET b$=""
20 INPUT LINE a$
30 IF a$<>"" THEN LET b$=b$+a$+CHR$ 254: GO TO 20
40 LET len=LEN b$
50 LET b$(len)=CHR$ 255
60 LET end=begin+len
70 FOR a=1 TO len
80 POKE (begin+a-1),CODE (b$(len-a+1))-32
90 NEXT a
100 PRINT "TEXT: ";begin;"...";end-1''"LENGTH=";len
```

Текст надо вводить по запросу программы по строкам (не более 32 символов). При вводе текст строки будет автоматически завершен символом перевода на новую строку. Нажатие ENTER без ввода текста означает завершение текста и ввод символа конца текста. После этого будет произведено кодирование и выданы следующие значения: начальный и конечный адрес расположения текста в памяти, а также длину текста. Первый и последний параметры можете использовать для сохранения "закодированного" текста на ленте или диске, а второй Вы должны будете подставить в строку 10 листинга кодового блока вместо nnnnn).

© Руслан Крылов, г. Владивосток, 1994.

Предлагаю Вам к разделу "Этюды" небольшую, но довольно эффектную программу, выполняющую очистку экрана. Подобной я у Вас пока не встречал. Программа очищает экран от центра к четырём вершинам, при этом, в центре экрана образуется чёрный ромб, увеличивающийся в размерах, пока им не будет заслонен (очищен) весь экран.

```
00010             ORG       40000
00020 ONE         LD        A,B
00030             CP        #C0
00040             RET       NC
00050             CALL      8881
00060             LD        B,#BF
00070             LD        A,C
00080             AND       #07
00090             ADD       A,#F1
00100             LD        C,A
00110             LD        A,(BC)
00120             AND       (HL)
00130             LD        (HL),A
00140             RET
00150 TWO         LD        A,#80
00160             ADD       A,E
00170             LD        C,A
00180             LD        A,#60
00190             ADD       A,D
00200             LD        B,A
00210             CALL      ONE
00220             LD        A,#80
00230             SUB       E
00240             LD        C,A
00250             LD        A,#60
00260             ADD       A,D
00270             LD        B,A
00280             CALL      ONE
00290             LD        A,#80
00300             SUB       E
00310             LD        C,A
00320             LD        A,#60
00330             SUB       D
00340             LD        B,A
00350             CALL      ONE
00360             LD        A,#80
00370             ADD       A,E
00380             LD        C,A
00390             LD        A,#60
00400             SUB       D
00410             LD        B,A
00420             CALL      ONE
00430             DEC       E
00440             INC       D
00450             LD        A,E
00460             CP        #FE
00470             RET       NC
00480             JR        TWO
00490             ENT
00500 START       LD        HL,22528
00510             LD        DE,22529
00520             LD        BC,767
00530             LD        (HL),1
00540             LDIR
00550             LD        DE,255
00560 THREE       INC       E
00570             INC       E
00580             PUSH      DE
00590             CALL      TWO
00600             POP       DE
00610             LD        A,E
```

```
00620             CP        #DF
00630             RET       Z
00640             JR        THREE
00650             RET
Start of object   : #9C40 (40000)
Length            : #0065 (00101)
Executes          : #9C87 (40071)
```

Сразу после старта программы (строка 490) происходит установка атрибутов для всего экрана, значение которых задается числовым параметром в строке 530 (в данном примере PAPER 0; INK 1, но Вы можете задать своё значение при помощи хотя бы POKE 40081,N). Затем начинается собственно формирование "ромба", очищающего экран.

Если Вы добавите строку:

```
00545             ENT
```

то будете иметь ромб без предварительной установки атрибутов. Иными словами, для этого надо запустить блок с адреса:

```
Executes          : #9C94 (40084)
```

Шестнадцатеричный дамп блока:

```
9C40 | 78  FE  C0  D0  CD  B1  22  0б | 88
9C48 | BF  79  E6  07  C6  P1  4F  0A | 19
9C50 | A6  77  C9  3E  80  83  4F  3E | A0
9C58 | 60  82  47  CD  40  9C  3E  80 | 84
9C60 | 93  4F  3E  60  82  47  CD  40 | 52
9C68 | 9C  3E  80  93  4F  3E  60  92 | 70
9C70 | 47  CD  40  9C  3E  80  83  4F | 8C
9C78 | 3E  60  92  47  CD  40  9C  1D | 51
9C80 | 14  7B  FE  FE  D0  18  CC  21 | 7C
9C88 | 00  58  11  01  58  01  FF  02 | E8
9C90 | 36  01  ED  B0  11  FF  00  1C | 2C
9C98 | 1C  D5  CD  53  9C  D1  7B  FE | 2B
9CA0 | DF  C8  18  F3  C9  00  00  00 | B7
```

© Андрей Нефедов, г. Новокузнецк, 1994.

Хочу поделиться с читателями некоторыми процедурами, взятыми из разных игровых и прикладных программ. Эти процедуры позволяют создавать разные звуковые эффекты, которые играют немаловажную роль в различных программах, создаваемых любителями сделать что-нибудь своими руками.

Процедуры эти довольно простые, как в написании, так и в обращении. Они, при различных значениях изменяемых параметров, помеченных в листингах как "var", помогут создать довольно обширный спектр звуковых эффектов.

Процедура 1.

```
00010             ORG       40000
00020             ENT
00030             LD        A,(23624)
00040             RRA
00050             RRA
00060             RRA
00070             LD        B,240     ;var
00080             LD        C,254
00090 LB          DEC       H
00100             JR        NZ,LA
00110             XOR       16
00120             OUT       (C),A
00130             LD        H,238     ;var
00140 LA          DEC       L
00150             JR        NZ,LB
00160             XOR       16
00170             OUT       (C),A
00180             LD        L,254     ;var
00190             DJNZ      LB
00200             RET
Start of object   : #9C40 (40000)
Length            : #001F (00031)
Executes          : #9C40 (40000)
 9C40 | 3A  48  5C  1F  1F  1F  06  F0 | 0D
 9C48 | 0E  FE  25  20  06  EE  10  ED | 26
```

```
                  9C50 | 79   26   EE   2D   20   F4   BE   10 | B8
                  9C58 | ED   79   2E   FE   10   EC   C9   00 | 4B
```

Процедура 2.

```
00010             ORG       40000
00020             ENT
00030             LD        BC,64004 ;var
00040             LD        HL,512   ;var
00050             LD        DE,2     ;var
00060 LA          PUSH      HL
00070             PUSH      DE
00080             PUSH      BC
00090             CALL      949
00100             POP       BC
00110             POP       DE
00120             POP       HL
00130             LD        A,L
00140             SUB       C
00150             LD        L,A
00160             DJNZ      LA
00170             RET
Start of object   : #9C40 (40000)
Length            : #0018 (00024)
Executes          : #9C40 (40000)
                  9C40 | 01   04   FA   21   00   02   11   02 | 11
                  9C48 | 00   E5   D5   C5   CD   B5   03   C1 | A9
                  9C50 | D1   E1   7D   91   6F   10   F2   C9 | E6
```

Процедура 3.

```
00010             ORG       40000
00020             ENT
00030             LD        A,(23624)
00040             SRL       A
00050             SRL       A
00060             LD        L,A
00070             LD        E,0       ;var
00080             LD        D,128     ;var
00090 LC          PUSH      DE
00100 LB          LD        B,E
00110 LA          DJNZ      LA
00120             LD        A,(BC)
00130             AND       248
00140             OR        L
00150             OUT       (254),A
00160             INC       C
001^0             DEC       D
00180             JR        NZ,LB
00190             POP       DE
00200             INC       E
00210             DEC       D
00220             JR        NZ,LC
00230             RET
Start of object   : #9C40 (40000)
Length            : #0020 (00032)
Executes          : #9C40 (40000)
                  9C40 | 3A   48   5C   CB   3F   CB   3F   6F | 3D
                  9C48 | 1E   00   16   80   D5   43   10   FE | BE
                  9C50 | 0A   E6   F8   B5   D3   FE   0C   15 | 7B
                  9C58 | 20   F3   D1   1C   15   20   ED   C9 | DF
```

Процедура 4.

```
00010             ORG       40000
00020             ENT
00030             LD        A,(21320)
00040             RRCA
00050             RRCA
00060             RRCA
```

```
00070             LD        E,200     ;var
00080 LB          DI
00090             OUT       (254),A
00100             XOR       17
00110             LD        B,E
00120 LA          DJNZ      LA
00130             DEC       E
00140             JR        NZ,LB
00150             EI
00160             RET
Start of object   : #9C40 (40000)
Length            : #0015 (00021)
Executes          : #9C40 (40000)
                    9C40 | 3A   48   53   0F   0F   0F   1E   C8 | C4
                    9C48 | F3   D3   FE   EE   11   43   10   FE | F8
                    9C50 | 1D   20   F5   FB   C9   00   00   00 | E2
```

Процедура 5.

```
00010             ORG       40000
00020             ENT
00030             LD        E,50      ;var
00040             LD        C,254
00050 LC          LD        D,A
00060             RES       4,D
00070             RES       3,D
00080             LD        B,E
00090 LB          CP        B
00100             JR        NZ,LA
00110             LD        D,24
00120 LA          OUT       (C),D
00130             DJNZ      LB
00140             DEC       A
00150             JR        NZ,LC
00160             RET
Start of object   : #9C40 (40000)
Length            : #0017 (00023)
Executes          : #9C40 (40000)
                    9C40 | 1E   32   0E   FE   57   CB   A2   CB | C7
                    9C48 | 9A   43   B8   20   02   16   18   ED | B6
                    9C50 | 51   10   F7   3D   20   EE   C9   00 | 58
```

Предоставим читателях возможность самим поэкспериментировать с изменяемыми числовыми параметрами и обратим внимание на другой момент. Так как порт звукового динамика и бордюра - один и тот же, то звуковые эффекты неизбежно влияют на цвет бордюра. Если Вам не надо, чтобы цвет его менялся, надо предусматривать специальные меры - блокировать 0…2 биты числа, выдаваемого в порт (#FE). Или же, наоборот, звуковой эффект может приводить к изменению цвета бордюра (0-2 биты), если того требует идея. Так, в процедуре 1 попробуйте изменить строки 110 и 160, задавая числовой параметр для XOR в пределах от 16 до 23, да ещё так, чтобы он не обязательно был одинаковым в этих строках.

В Процедуре 2 используется стандартный BEEPER из ПЗУ, поэтому цвел бордюра будет определяться содержимым системной ячейки 23624 (BORDCR).

Процедура 3 тоже использует значение BORDCR. От её значения будет зависеть цвет бордюра в момент звукового сигнала. Например, при белом бордюре, он станем жёлтым, при жёлтом - изменится на зелёный и т.д. Если Вы измените строку 130, например, так:

```
00130             XOR       255
```

то при изначально зеленом бордюре в момент звукового сигнала получите полосы всех цветов.

Не менее интересные результаты дадут эксперименты с процедурой 4. Здесь вообще в качестве переменной используется ячейка дисплейного файла. Видимо, в оригинале процедуры цвет бордюра в момент звукового сигнала зависел от того, что в этот момент находилось в этом месте экрана. Можете переделать программу:

```
00030             LD        A,N       ;var
```

а строки 40-60 - удалите (это, кстати, сократит процедуру на 4 байта). Попробуйте менять параметр N в диапазоне 0…7, меняя также при этом и параметр в строке 100 от 16 до 23 - Вы получите различные цвета полос на бордюре в момент звукового сигнала.

А в процедуре 5 в момент звукового сигнала Вы увидите все цвета радуги на экране.

Андрей тоже просит напечатать его адрес - он надеется, что вместе с новыми друзьями они смогут разгадать какие-нибудь секреты ZX-Spectrum:

654006, Кемеровская обл., г. Новокузнецк, ул. Филиппова, 16-9,
Нефедов Андрей.

© Вячеслав Пономарев, г. Черногорск (Хакасия), 1994.

Предлагаю Вам программу, которую написал под впечатлением от многочисленных "музыкалок". Эта программа выводит любой текст произвольным (загруженным) символьным набором. Особенность программы в том, что текст выводится в нижней трети экрана и буквы по высоте равны одной трети экрана. Вся хитрость в том, что буквы рисуются не в дисплейном файле, а в файле атрибутов, цветом PAPER и INK, то есть Вы можете нарисовать в нижней трети любой рисунок любым цветом INK и буквы будут "бежать" по рисунку, а рисунок как бы просвечиваться через буквы.

```
00010             ORG       40000
00020             ENT
00030 BEGIN       LD        HL,ADR
00040             LD        (TEXT),HL
00050 MN          LD        DE,(23606)
00060             LD        HL,(TEXT)
00070             LD        A,(HL)
00080             CP        13
00090             JR        Z,BEGIN
00100             CP        16
00110             JP        Z,L1
00120             CP        17
00130             JP        Z,L2
00140             LD        H,0
00150             LD        L,A
00160             ADD       HL,HL
00170             ADD       HL,HL
00180             ADD       HL,HL
00190             ADD       HL,DE
00200             LD        BC,8
00210             LD        DE,BUFFER
00220             LDIR
00230             LD        B,8
00240 M3          PUSH      BC
00250 PAUSE       LD        B,1
00260 MM          HALT
00270             DJNZ      MM
00280             LD        DE,#5A1F
00290             LD        B,8
00300             LD        HL,BUFFER
00310             LD        (POINTB),HL
00320 M1          LD        HL,(POINTB)
00330             LD        A,(HL)
00340             SLA       A
00350             JR        C,MSET
00360             LD        A,(DE)
00370             AND       #C7
00380 PAPER       OR        #20
00390             LD        (DE),A
00400 M0          LD        HL,#20
00410             ADD       HL,DE
00420             EX        DE,HL
00430             LD        HL,(POINTB)
00440             INC       HL
00450             LD        (POINTB),HL
00460             DJNZ      M1
00470             LD        HL,BUFFER
00480             LD        B,8
00490 M2          LD        A,(HL)
00500             SLA       A
00510             LD        (HL),A
00520             INC       HL
00530             DJNZ      M2
00540             LD        HL,#5A00
```

```
00550             LD        C,8
00560 NEXTL       LD        B,31
00570 NEXTC       INC       HL
00580             LD        A,(HL)
00590             AND       #38
00600             LD        E,A
00610             DEC       HL
00620             LD        A,(HL)
00630             AND       #C7
00640             OR        E
00650             LD        (HL),A
00660             INC       HL
00670             DJNZ      NEXTC
00680             INC       HL
00690             DEC       C
00700             JR        NZ,NEXTL
00710             POP       BC
00720             DJNZ      M3
00730 PROD        LD        HL,(TEXT)
00740             INC       HL
00750             LD        (TEXT),HL
00760             LD        A,#7F
00770             IN        A,(#FE)
00780             AND       1
00790             RET       Z
00800             JP        MN
00810 MSET        LD        A,(DE)
00820             AND       #C7
00830 INK         OR        #10
00840             LD        (DE),A
00850             JR        M0
00860 L1          CALL      LLL
00870             LD        (INK+1),A
00880             JR        PROD
00890 L2          CALL      LLL
00900             LD        (PAPER+1),A
00910             JR        PROD
00920 LLL         LD        HL,(TEXT)
00930             INC       HL
00940             LD        (TEXT),HL
00950             LD        A,(HL)
00960             LD        HL,POICOL
00970             LD        D,0
00980             LD        E,A
00990             ADD       HL,DE
01000             LD        A,(HL)
01010             RET
01020 BUFFER      DEFW      0,0,0,0
01030 TEXT        DEFW      0
01040 POINTB      DEFW      0
01050 POICOL      DEFB      0,8,16,24
01060             DEFB      32,40,48,56
01080 ADR         DEFB      16,2,17,5
01090             DEFM      " HELLO, "
01100             DEFB      16,7,17,2
01110             DEFM      "ALL USERS"
01120             DEFB      16,2,17,5
01130             DEFM      " !!! "
01140             DEFB      17,1
01150             DEFM      "    "
01160             DEFB      13
Start of object   : #9C40 (40000)
Length            : #00F9 (00249)
Executes          : #9C40 (40000)
```

В строке 30 задается адрес текста в ОЗУ. В строке 40 его значение сохраняется в переменной TEXT. В строке 50 задается значение CHARS того символьного набора, которым должна выполняться печать. Далее анализируются символы текста. В строке 80 проверяется признак конца текста. Если он зафиксирован, программа зацикливается и текст печатается сначала. В строке 100 анализируется код управления цветом INK, а в строке 120 - код управления PAPER.

Строки 140-190 находят образ символа по его коду, а 200-220 переносят восемь его байтов в служебный буфер.

В строках 250-270 организована задержка, определяющая скорость движения бегущей строки, она зависит от числового параметра, задаваемого в регистре B. Однако при других значениях B слишком заметными становятся рывки при перемещении букв.

В строке 540 задается адрес начала нижней трети экрана в файле атрибутов. В строках 550-700 происходит сдвиг атрибутов (PAPER) нижней трети - влево.

В строках 760-790 выполняется опрос клавиши SPACE, если она нажата, то происходит выход из программы.

Текст кодируется - как обычно, но в конце текста обязательно должен стоять маркер CHR$ 13. Чтобы сменить цвет букв или фона (не забывайте, что цвет букв PAPER!), нужно задать управляющие коды CHR$ 16; CHR$… - для цвета INK (фон букв) или CHR$ 17; CHR$… - для цвета PAPER (цвет букв).

А теперь шестнадцатеричный дамп:

9C40	21	10	9D	22	04	9D	ED	5B	B5
9C48	36	5C	2A	04	9D	7E	FE	0D	CA
9C50	28	EE	FE	10	CA	DB	9C	FE	4F
9C58	11	CA	E3	9C	26	00	6F	29	0C
9C60	29	29	19	01	08	00	11	FC	7D
9C68	9C	ED	B0	06	08	C5	06	01	17
9C70	76	10	FD	11	1F	5A	06	08	27
9C78	21	FC	9C	22	06	9D	2A	06	C2
9C80	9D	7E	CB	27	38	4D	1A	E6	AE
9C88	C7	F6	20	12	21	20	00	19	6D
9C90	EB	2A	06	9D	23	22	06	9D	CC
9C98	10	E4	21	FC	9C	06	08	7E	6D
9CA0	CB	27	77	23	10	F9	21	00	F2
9CA8	5A	0E	08	06	1F	23	7E	E6	60
9CB0	38	5F	2B	7E	E6	C7	B3	77	63
9CB8	23	10	F2	23	0D	20	EC	C1	76
9CC0	10	AB	2A	04	9D	23	22	04	2B
9CC8	9D	3E	7F	DB	FE	E6	01	C8	46
9CD0	C3	46	9C	1A	E6	C7	F6	10	DE
9CD8	12	18	B1	CD	EB	9C	32	D7	AC
9CE0	9C	18	DF	CD	EB	9C	32	8A	1F
9CE8	9C	18	D7	2A	04	9D	23	22	1F
9CF0	04	9D	7E	21	08	9D	16	00	87
9CF8	5F	19	7E	C9	00	00	00	00	53
9D00	00	00	00	00	00	00	00	00	9D
9D08	00	08	10	18	20	28	30	38	85
9D10	10	02	11	05	20	48	45	4C	CE
9D18	4C	4F	2C	20	10	07	11	02	C6
9D20	41	4C	4C	20	55	53	45	52	F5
9D28	53	10	02	11	05	20	21	21	A2
9D30	21	20	11	01	20	20	20	20	A0
9D38	0D	00	00	00	00	00	00	00	E2

© Сергей Гуляев, республика Коми, 1994 г.

Очень люблю Ваш новый раздел ЭТЮДЫ. Решив поддержать идею А.А.Иванова из г. Печора, я высылаю Вам две программы.

Программа 1. Действие её таково: от центра экрана к его краям расходятся разноцветные полосы.

```
00010             ORG       40000
00020             ENT
00030             LD        IX,#000F
00040             LD        HL,#0010
00050             LD        B,#10
00060 L1          PUSH      BC
00070             PUSH      HL
00080             CALL      L2
00090             PUSH      IX
00100             POP       HL
```

```
00110             CALL      L2
00120             POP       HL
00130             POP       BC
00140             INC       HL
00150             DEC       IX
00160             LD        A,#FF
00170             LD        (L11),A
00180             DJNZ      L1
00190             RET
00200 L2          LD        B,#18
00210             LD        A,(L10)
00220             ADD       A,2
00230             LD        (L10),A
00240 L3          PUSH      BC
00250             PUSH      HL
00260             CALL      L4
00270             POP       HL
00280             LD        A,(L10)
00290             AND       #7F
00300             LD        C,A
00310             LD        A,#20
00320             CALL      L6
00330             LD        DE,#0020
00340             ADD       HL,DE
00350             POP       BC
00360             DJNZ      L3
00370             RET
00380 L4          LD        A,(L11)
00390             CP        0
00400             RET       Z
00410             DEC       A
00420             LD        (L11),A
00430             LD        C,A
00440             LD        B,A
00450             LD        A,#10
00460 L5          OUT       (#FE),A
00470             DJNZ      L5
00480             LD        B,C
00490             XOR       A
00500             OUT       (#FE),A
00510             RET
00520 L6          PUSH      HL
00530             PUSH      AF
00540             EX        DE,HL
00550             LD        HL,#5800
00560             ADD       HL,DE
00570             LD        (HL),C
00580             LD        HL,#4000
00590             LD        A,D
00600             OR        A
00610             JR        Z,L7
00620             LD        H,#47
00630 L7          CP        2
00640             JR        NZ,L8
00650             LD        H,#4E
00660 L8          ADD       HL,DE
00670             POP       AF
00680             PUSH      HL
00690             LD        L,A
00700             LD        H,0
00710             ADD       HL,HL
00720             ADD       HL,HL
00730             ADD       HL,HL
00740             EX        DE,HL
```

```
00750             LD        HL,(#5C36)
00760             ADD       HL,DE
00770             EX        DE,HL
00780             POP       HL
00790             LD        B,8
00800 L9          LD        A,(DE)
00810             INC       DE
00820             LD        (HL),A
00830             INC       H
00840             DJNZ      L9
00850             POP       HL
00860             RET
00870 L10         DEFB      0
00880 L11         DEFB      0
Start of object   : #9C40 (40000)
Length            : #008E (00142)
Executes          : #9C40 (40000)
```

Эффект происходит со звуковым сопровождением.

Мы хотим обратить вникание читателей на строку 310 листинга: здесь задается код ПРОБЕЛА, печать которого разными цветами и создает эффект цветных полос. Значение CHARS символьного набора для печати задается в строке 750.

Шестнадцатеричный дамп процедуры:

```
9C40 | DD   21   0F   00   21   10   00   06 | 20
9C48 | 10   C5   E5   CD   61   9C   DD   E5 | 2A
9C50 | E1   CD   61   9C   E1   C1   23   DD | 39
9C58 | 2B   3E   FF   32   CD   9C   10   E9 | F0
9C60 | C9   06   18   3A   CC   9C   C6   02 | 4D
9C68 | 32   CC   9C   C5   E5   CD   84   9C | 35
9C70 | E1   3A   CC   9C   E6   7F   4F   3E | 81
9C78 | 20   CD   9B   9C   11   20   00   19 | 82
9C80 | C1   10   E8   C9   3A   CD   9C   FB | 3F
9C88 | 00   C8   3D   32   CD   9C   4F   47 | 5A
9C90 | 3E   10   D3   FE   10   FC   41   AF | 47
9C98 | D3   FE   C9   E5   F5   EB   21   00 | B4
9CA0 | 58   19   71   21   00   40   7A   B7 | B0
9CA8 | 28   02   26   47   FE   02   20   02 | FD
9CB0 | 26   4E   19   F1   E5   6F   26   00 | 44
9CB8 | 29   29   29   EB   2A   36   5C   19 | 8F
9CC0 | EB   E1   06   08   1A   13   77   24 | FE
9CC8 | 10   FA   E1   C9   00   00   00   00 | 18
```

Программа 2. Она является разновидностью программы "Гашение экрана".

Эта программа аналогична уже опубликованной нами (в РЕВЮ-94 № 3, стр. 45) процедуре, имитирующей действие FADE_0) MEGABASIC. Отличие предлагаемой сейчас в том, что происходящее на экране сопровождается звуковым эффектом.

```
00010             ORG       40000
00020             ENT
00030             LD        HL,LABEL1
00040             ADD       A,(HL)
00050             LD        (LABEL1),A
00060             LD        A,R
00070             AND       #7F
00080             ADD       A,(HL)
00090             LD        (LABEL1),A
00100             LD        HL,(LABEL3)
00110             INC       HL
00120             LD        A,H
00130             AND       #03
00140             LD        H,A
00150             LD        (LABEL3),HL
00160             LD        A,(HL)
00170             LD        HL,LABEL1
00180             ADD       A,(HL)
00190             LD        (LABEL1),A
00200             LD        L,A
```

```
00210             LD        H,0
00220             ADD       HL,HL
00230             ADD       HL,HL
00240             LD        DE,#5800
00250             LD        BC,#0300
00260             LDIR
00270 L1          LD        HL,#5800
00280             LD        D,0
00290             LD        BC,#0300
00300 L2          LD        A,(HL)
00310             CP        0
00320             JR        Z,L3
00330             LD        D,#FF
00340             DEC       (HL)
00350             LD        A,#10
00360             OUT       (#FE),A
00370             XOR       A
00380             OUT       (#FE),A
00390 L3          DEC       BC
00400             INC       HL
00410             LD        A,B
00420             OR        C
00430             JR        NZ,L2
00440             CP        D
00450             JR        NZ,L1
00460             LD        HL,#4000
00470             LD        DE,#4001
00480             LD        (HL),0
00490             LD        BC,#1800
00500             LDIR
00510             RET
00520 LABEL1      DEFB      0
00530 LABEL2      DEFB      0
00540 LABEL3      DEFB      0
00550 LABEL4      DEFB      0
Start of object   : #9C40 (40000)
Length            : #0061 (00097)
Executes          : #9C40 (40000)
```

Шестнадцатеричный дамп:

```
9C40 | 21  9D  9C  86  32  9D  9C  ED | 14
9C48 | 5F  E6  7F  86  32  9D  9C  2A | C3
9C50 | 9F  9C  23  7C  E6  03  67  22 | 38
9C58 | 9F  9C  7E  21  9D  9C  86  32 | BF
9C60 | 9D  9C  6F  26  00  29  29  11 | 2D
9C68 | 00  58  01  00  03  ED  B0  21 | 1E
9C70 | 00  58  16  00  01  00  03  7E | FC
9C78 | FE  00  28  0A  16  FF  35  3E | CC
9C80 | 10  D3  FE  AF  D3  FE  0B  23 | AB
9C88 | 78  B1  20  EB  BA  20  E0  21 | 33
9C90 | 00  40  11  01  40  36  00  01 | F5
9C98 | 00  18  ED  B0  C9  00  00  00 | B2
9CA0 | 00  00  00  00  00  00  00  00 | 3C
```

© Евгений Любимов, г. Новокузнецк, 1994.

Хочу предложить в рубрику "Этюды" программу, действие которой аналогично команде Мега-Бейсика FADE_0. Она релоцируемая, может загружаться в любой адрес и стартовать с адреса загрузки.

```
00010             ORG       40000
00020             ENT
00030             LD        BC,768
00040             LD        DE,22528
00050             LD        HL,768
00060             LDIR
00070             LD        B,255
00080 L3          PUSH      BC
```

```
00090             LD        HL,22528
00100             LD        BC,768
00110 L2          LD        A,(HL)
00120             CP        1
00130             CALL      NC,L1
00140             INC       HL
00150             DEC       BC
00160             LD        A,B
00170             OR        C
00180             JR        NZ,L2
00190             POP       BC
00200             DJNZ      L3
00210             RET
00220 L1          DEC       A
00230             LD        (HL),A
00240             RET
Start of object   : #9C40 (40000)
Length            : #0027 (00039)
Executes          : #9C40 (40000)
                    9C40 | 01  00   03   11   00   58   21   00 | 6A
                    9C48 | 03  ED   B0   06   FF   C5   21   00 | 6F
                    9C50 | 58  01   00   03   7E   FE   01   D4 | 99
                    9C58 | 64  9C   23   0B   78   B1   20   F4 | 5F
                    9C60 | C1  10   EA   C9   3D   77   C9   00 | FD
```

Эта процедура по эффекту похожа на аналогичную, предложенную в РЕВЮ-94 № 3, хотя, справедливости ради, следует заметить, что эффект всё же отличается.

© Алексей Прокопенко, г. Ростов-на-Дону, 1994.

Предлагаю Вам свою программу для скроллинга всего экрана внутри знакомест.

```
00010             ORG       40000
00020             ENT
00030             LD        HL,16384
00040             LD        BC,6144
00050 LOOP        LD        A,(HL)
00060             RLCA
00070             LD        (HL),A
00080             INC       HL
00090             DEC       BC
00100             LD        A,B
00110             OR        C
00120             JR        NZ,LOOP
00130             RET
Start of object   : #9C40 (40000)
Length            : #0010 (00016)
Executes          : #9C40 (40000)
```

В строке 60 может стоять любая циклическая или нециклическая команда сдвига регистра A. В зависимости от этого можно получать различные эффекты.

Например, при использовании команда SRL A или SLA A получим интересную очистку экрана, а при использовании RRCA или RLCA - интересный эффект с экраном. Для этого добавьте строки:

```
00022             LD        B,24
00024 L1          PUSH      BC

00122             POP       BC
00124             DJNZ      L1
```

Получим вариант программы, "вращающей" изображение внутри каждого знакоместа экрана три раза. Параметры блока кодов станут при этом следующими:

```
Start of object   : #9C40 (40000)
Length            : #0016 (00022)
Executes          : #9C40 (40000)
                    9C40 | 06  18   C5   21   00   40   01   00 | 21
                    9C48 | 18  7E   07   77   23   0B   78   B1 | 4F
                    9C50 | 20  F7   C1   10   ED   C9   00   00 | 8A
```

© Руслан Пак, г. Климовск Тульской области, 1994.

Я испытал программу "ВОТ ТАК!", опубликованную в РЕВЮ-94 № 3 (стр.47). Картинку она проявляет

хорошо, а вот атрибутов на ней нет. Предлагаю усовершенствование этой программы.

```
00010              ORG       50000
00020 W1           LD        F,1
00030              AND       A
00040              EX        AF,AF'
00050              LD        B,9
00060 W2           LD        DE,#1800
00070              LD        HL,#DF38
00080 W3           DEC       HL
00090              DEC       DE
00100              LD        A,D
00110              OR        E
00120              JR        Z,W4
00130              SET       6,D
00140              EX        AF,AF'
00150              RLA
00160              RLA
00170              RLA
00180              RLA
00190              PUSH      AF
00200              AND       (HL)
00210              EX        DE,HL
00220              OR        (HL)
00230              EX        DE,HL
00240              LD        (DE),A
00250              POP       AF
00260              EX        AF,AF'
00270              RES       6,D
00280              JR        W3
00290 W4           DJNZ      W2
00300              RET
00310              ENT
00320              LD        DE,#5800
00330              LD        HL,#DF38
00340              LD        BC,#0300
00350              LDIR
00360              JR        W1
Start of object    : #C350 (50000)
Length             : #0035 (00053)
Executes           : #C378 (50040)
```

Загрузите заставку под адрес 51000, затем запустите программу с адреса 50040.

Шестнадцатеричный дамп процедуры:

C350	3E	01	A7	08	06	09	11	00	21
C358	18	21	38	DF	2B	1B	7A	B3	DE
C360	28	13	CB	F2	08	17	17	17	68
C368	17	F5	A6	EB	B6	EB	12	F1	6C
C370	08	CB	B2	18	E7	10	DF	C9	6F
C378	11	00	58	21	38	DF	01	00	DD
C380	03	ED	B0	18	CB	00	00	00	C6

* * *

"Хаккер-93"

ИФК: В своих письмах многие читатели задают вопрос, который интересует, наверное, не только их: "Где же он, "Хаккер-93", автор которого В. Корчагин из Нижнего Тагила"? Письмо В. Корчагина так взбудоражило читателей, что Мы до сих пор получаем по этому поводу письма полярно противоположных мнений. Так, например, один из читателей спрашивает: "Как понравится "Хаккеру-93" присутствие в моём загрузчике конвейера команд плюс проверка контрольной суммы подпрограммы, находящейся по адресу 102 (#0066) в ПЗУ? Потом я всё это закодирую и опять заставлю хакера "копать землю".

Друзья. Не переходите, пожалуйста, разумные пределы. Больше руководствуйтесь здравым смыслом, а то можете дойти до абсурда. Если Вы создаете программу только ради того, чтобы она оказалась "не по зубам" "Хаккеру-93", то, пожалуйста, дерзайте. Но захотят ли приобрести Вашу программу тысячи добропорядочных пользователей Spectrum-совместимых компьютеров с модифицированными ПЗУ, у которых тоже не сойдется контрольная сумма? Видоизмененные ПЗУ стали объективной реальностью и разработчики программ должны это учитывать, если рассчитывают на массовую реализацию результатов своего труда. Не защищаться от ПЗУ надо, а наоборот, стремиться к тому, чтобы программа работала с любым ПЗУ. Если она определяет наличие 128К или музыкального процессора, то необходимо добиваться, чтобы она делала это корректно на компьютерах любой модели с любым ПЗУ и т.д. А то Вы просто останетесь в одиночестве со своей программой. Либо появятся предложения типа: "Для того чтобы программа заработала на компьютере такой-то модели необходимо переделать её следующим образом…" - то есть "заяц всё равно погибает". А те, которые сами не сумеют переделать, пойдут на рынок и охотнее приобретут Вашу программу у взломавшего её хакера, чем у Вас. Поэтому лучше направьте силы и умение на более полезный для окружающих труд - и Вы получите гораздо большее моральное и материальное удовлетворение!

Недавно мы получили посылку от Владимира Анатольевича Корчагина, в которой он прислал нам для исследования прошивки с разработанным им монитором "Хаккер-93", а также полнейшую документацию к нему. Теперь можем сообщить нашим читателям новость, которую по достоинству оценить сможет только истинный поклонник Speccy. Владимир Анатольевич, в качестве Новогоднего подарка всем любителям ZX-Spectrum, дал свое согласие на публикацию этих материалов, так что скоро с ними смогут ознакомиться все желающие. Мысль остаться в одиночестве с результатами своего труда явно не устраивает разработчика "Хаккер-93". Мы уверены, что любители ZX будут очень признательны Владимиру Анатольевичу за такое решение. Полная документация на "Хаккер-93" войдет в сборник "ZX-FORUM-2", так что следите за нашим рекламным листком.

Теперь каждый сможет на деле попробовать, что же это такое "Хаккер-93". Мы же можем сказать следующее. Как-то давным-давно, мы пытались взломать загрузчик (типа ALKATRAZ) программы ADVANCED PINBALL. Мы делали тогда только первые шаги в этом направлении, да и методика взлома не была на таком уровне, как сегодня. Тогда мы так и не довели это дело до конца… А сейчас решили проверить "Хаккер-93" в деле - и проверили (кстати, методом теневого ОЗУ). Адаптация этой игры под диск (с взломом загрузчика) заняла примерно полчаса.

Кстати, "Хаккер-93" предлагается автором не как некое "абсолютное средство для взломщика", а как дополнительные сервисные возможности, которые получают пользователи компьютера, в основном, начинающие, так как опытный хакер и без него расколет любой ALKATRAZ. И что особо ценно, все эти возможности до деталей описаны в документации.

****

"Старые" программы.

ИФК: Степанищев Е.В. из г. Лениногорска (Татарстан) пишет:

КОРР: Недавно обнаружил неприятную особенность MONS-3: попробуйте выполнить в пошаговом режиме команду HALT - и он зависнет.

ИФК: Это, действительно, так - мы проверили. Однако, в MONS-4 -всё в порядке. А вот, что, в связи с этим, категорически утверждает наш читатель Игорь Назаров из Нижнего Новгорода.

КОРР: Никогда не пользуйтесь MONS'ами и GENS'ами версий ниже 4 - сэкономите байты - потеряете нервы.

ИФК: Несколько слов скажем о MONS'ах и GENS'ах. В ответ на положительные отзывы одних наших читателей по поводу пакета DEVPAC, мы получили несколько сердитых писем о том, что мы хвалим программы, которые "отжили свой век", и вообще, это "откровенное убожество" и так далее.

Однако позволим себе пару слов в защиту программ, которые, несмотря на свои недостатки, стали уже "классикой" - даже самые "сердитые" читатели начинали именно с них.

Мы понимаем, что творческая мысль не стоит на месте. Практика показала, что в последнее время на отечественном рынке появляются программные средства, ничуть не уступающие лучшим западным образцам. Так, в письмах читатели очень восторженно отзываются о сравнительно новом ассемблере TASM 128. Пользуясь случаен,

передаем огромную благодарность наших читателей его автору (авторам?) "RST7". Кстати, текст программы и коды самого TASM размещаются в дополнительных страницах ОЗУ 128К машины, за исключением нескольких байтов в первой строке Бейсик-загрузчика. Ассемблер свободно работает с диском, в том числе, предусматривает при компиляции динамическую подгрузку текстов с диска.

Но вернёмся на грешную землю. Дисковод, ОЗУ 128К - хорошо, если есть возможность… Для многих ещё долго это будет голубой мечтой - не все по теперешним временам могут позволить себе даже годовую подписку на ZX-РЕВЮ - мы получаем такие письма. Подавляющее большинство пользователей в стране имеют обычный ZX-48K с магнитофоном! Они не могут воспользоваться ни преимуществами TASM 128, ни IS-ASSEMBLER'ом в системе IS-DOS. Но заниматься программированием мечтают ничуть не меньше!

Мы никогда не утверждали, как нас неверно цитируют в письмах, что "пока ничего не придумано **удобнее** GENS'a" - в этом нас упрекают "сердитые" читатели. Как раз наоборот. Что касается **удобства** в управлении, то и MONS и GENS оставляют желать лучшего. Мы утверждали и утверждаем, что нам нравятся **потенциальные возможности** пакета MONS-GENS. Именно **пакета**! Речь идет о двустороннем транслировании исходного текста в объектный код **и наоборот**. Почему-то многие любят обходиться и без второй части этого процесса, предпочитая вручную набирать исходные тексты для всех программ. Мы считаем, что у программиста должна быть полная свобода действий: он должен иметь возможность не только набрать текст на ассемблере, чтобы затем получить из него объектный код, но и взять любой работающий блок кодов из любой "живой" программы и получить его исходный текст на ассемблере. Пакет MONS-GENS, в отличие от подавляющего большинства других программных пакетов, дает возможность двустороннего процесса. А вот что касается удобства в работе… Здесь, видимо, авторы сердитых писем имеют в виду прежде всего громоздкость редактирования текста в ассемблере GENS. Но Вы можете, вместо строчного редактора GENS, о удобством воспользоваться, например, текстовым редактором ZX-WORD Владимира Елисеева, и набирать исходный текст в нём, а затем в один момент конвертировать его в формат GENS'a (или обратно) утилитой, прилагаемой к пакету ZX-WORD. Очень скоро Вы сможете воспользоваться для набора текстов для GENS новым мощным текстовым редактором SPECCY WRITE Бориса Курицына - мы начинаем его дистрибуцию.

К слову, о двустороннем конвертировании, которое позволяет пакет MONS-GENS. Многие читатели недоумевают по поводу того, что мы тратим место на шестнадцатеричные дампы блоков кодов, листинги которых приводим на ассемблере: "Неужели ещё есть любители вводить программу покодово? Ведь на дворе не 1989-й год и любой желающий мог бы найти хоть какой-нибудь ассемблер". Обратите внимание: не пакет ассемблер-дизассемблер!

Дампы мы печатаем в дополнение к листингам на ассемблере по нескольким причинам. Во-первых, для дублирования информации. Несмотря на то, что практически все листинги в ZX-РЕВЮ "снимаются" с "живых" программ, Вы понимаете, что при подготовке материала к печати возможны всякого рода ошибки, а также дефекты при печати в типографии. Имея и листинг, и дамп, Вам легче будет выяснить, почему программа "не пошла".

Во-вторых, можем заверить, что "любители вводить программы покодово" существуют, и их достаточное количество. Недаром такой живой интерес вызвала программа MEMORY EDITOR Александра Гмарь. Кстати, попробуйте сосчитать, во сколько нажатий клавиш Вам обойдется набор какой-нибудь программы на ассемблере и с помощью MEMORY EDITOR. Во втором случае результат будет примерно в шесть (!) раз меньше. Во столько же раз меньше будут и затраты времени. Однако после набора объектного кода Вам ничто не помешает получить его исходный текст на ассемблере в формате GENS при помощи монитора-дизассемблера MONS.

К сказанному можем добавить, что нам очень понравился в работе ассемблер "ZEUS +", версии V3.1, которую доработал Сергей Ханцис из Ижевска (поставляется в составе дистрибутивной дискеты "Утилиты-2"). Можем сказать, что это самая лучшая полностью дисковая версия ZEUS из тех, которые нам приходилось встречать. Он имеет 14 новых команд. Среди прочих, есть возможность догружать текст с диска для слияния нескольких листингов программ в один, а также возможность ассемблирования программы под любой адрес прямо на диск - нет больше проблем со свободной памятью. В комплект пакета входит и утилита для конвертирования текстов программ на ассемблере из формата GENS в формат ZEUS.

Ещё пара слов о "старых" программах. Знаете ли Вы, что пакет EDITAS (EDITOR/ASSEMBLER) прекрасно понимает недокументированные команды с половинками индексных регистров IX и IY? Их необходимо обозначать так: XH и YH -старший (High) байт регистров IX и IY, а XL и YL - младший (Low) байт регистров IX и IY. Тогда Вы можете пользоваться такими командами, как INC XH, DEC YL, LD A,XL LD YH,D, OR XH, CP YL, ADD XL, SUB YH и многими другими.

****

ИФК: Феськов Кузьма из г. Абакан интересуется проблемами взлома программ.

КОРР: Пишу Вам по поводу статьи в ФОРУМе за этот год № 3. Она была посвящена взлому загрузчиков программ Bill Gilbert (см. стр. 52) путём подмены заставок. Кстати, этот метод помогает взломать защиту типа ALKATRAZ и загрузчики с недокументированными командами.

Всё дело в том, что после выгрузки блока кодов, как это было предложено в указанной статье, в отгруженном блоке не только может не оказаться взламываемого загрузчика, но даже, если он попал в этот блок, его очень трудно найти в большом пространстве, а это дополнительная трата времени. Я переработал эту программу, и вот, что у меня получилось.

```
10      EI
20      LD      BC,0
30      CALL    7997
```

```
40      POP     IX
50      LD      DE,1000
60      LD      A,#FF
70      CALL    1218
```

В этой программе всё основывается на том, что при вызове декомпрессора заставки командой CALL загрузчика, на стек заносится адрес возврата. В строках 10-30 задается пауза до нажатия любой клавиши. Строка снимает со стека адрес возврата. Если Вы хотите просмотреть весь загрузчик, то отнимите от этого числа примерно 500 байтов - чаще всего, этого достаточно. Затем, в строке 50 в DE заносим число выгружаемых байтов. И в строке 70 вызываем программу записи блока кодов, в котором содержится взламываемый загрузчик. Этот способ позволяет сразу найти загрузчик и не терять зря времени, "копая землю".

С удовольствием стал бы переписываться с кем-либо из синклеристов. Мой адрес:

662613, г. Абакан, ул. Ленинского Комсомола, д. 3, кв. 16,
Феськов Кузьма.

****

ИФК: Длинное письмо пришло от Шалыминова СП. из Нижнего Новгорода.

КОРР: Я уже писал Вам - Вы опубликовали моё объявление в РЕВЮ-93, № 11-12. Тогда я предлагал карты лабиринтов к играм. Пришло множество писем. На некоторые я ответил, выслал то, что просили, а некоторым я приношу извинения - пока я не в состоянии им ответить из-за большого объёма работы. Благодарю всех, кто прислал мне ту или иную информацию.

Спасибо за новую рубрику "Этюды". Я мог бы сказать ещё много хорошего в Ваш адрес, но я пишу по делу.

Во-первых, по поводу защиты программ. Я тоже часто занимаюсь дисковизацией игр и систем, поэтому мне, конечно, приходиться сталкиваться с такими вопросами. Так вот, я обычно, при проходе "ксорок" пользуюсь монитором "MONS-4". Технология проста - командой переноса блока кодов переношу "расксорку" в новую область памяти, где её "не тронут" - например, в буфер принтера. Далее, корректирую адреса переходов, ставлю точку останова - и вперёд (JP). Таким образом, я прохожу иногда до 100 и более ксорок. Кстати, новую ксорку, как правило, переносить не надо, лишь подправляем константы. Вы спросите, зачем всё это надо, если есть декодер, который Вы опубликовали? Отвечу. Около года назад я снимал защиту с какой-то игрушки, уже не помню, с какой. Там ксорки чередовались. То с регистром R, то с половинками IX и IY, то просто с числом - когда как. Я не знаю, какой маньяк это всё кодировал, но я прошел всё. (Где-то часа за два).

Сложных защит, как в письме А.А. Данилова (ZX-РЕВЮ-94, № 3, стр.49), с использованием IM-2, не видел. Недокументированные команды? Были. Но ведь я не в пошаговом режиме работаю! Так что, всё всегда получалось.

Далее, по поводу 128к. Как получить дизассемблер ПЗУ-0? (тот же номер РЕВЮ, стр. 54). Очень просто! Делаем в Бейсике CLEAR 49151, далее запускаем из экранной области (или откуда-нибудь ещё, если установить RAMTOP ниже) следующую программу:

```
DI
LD      HL,0
LD      DE,#C000
LD      BC,#7FFD
LD      A,0
OUT     (C),A
LD      BC,16384
LDIR
LD      BC,#7FFD
LD      A,16
OUT     (C),A
EI
RET
```

После возврата в BASIC, Вы будете с адреса 49152 иметь копию ПЗУ-0. Далее ее можно, например, выгрузить:

```
SAVE "ROM-0" CODE 49152,16384
```

При дизассемблировании нужно учитывать, что адреса смещены вместо #0042, например, будет #C042, вместо #3D0F получится #FD0F и тому подобное. Таким образом, я уже получил дизассемблер своего ПЗУ-0.

Кстати, у меня версия ПЗУ-128 за 1994 год. Там основное меню имеет вид:

```
128 TR-DOS
128 BASIC
48 TR-DOS
48 BASIC
CALCULATOR
```

Таким образом, все проблемы письма Н.А. Авдошкина решены. Мой совет - найти такое ПЗУ.

Далее, по поводу загрузчика из письма Е. Крюкова (ZX-РЕВЮ-94, № 3, стр. 59). У него неправильно определяется тип компьютера - 48к или 128к. То, что происходит в загрузчике, должно делаться не из BASIC, а из машинного кода. Делается это так:

С адреса 49152 (#CD00) переключается страницы (в 128 режиме). Команда OUT 32765,16 переключает туда 0-

ую страницу ОЗУ. Далее, по адресу 49152 пишется байт, например - "30". Потом OUT 32765,19 переключает третью страницу. Проверяется наличие этого байта (30) по тому же адресу. Если он там есть, значит, переключение не произошло - это 48к компьютер. Если же его нет, то это 128к, и надо грузить последний блок.

Но из BASIC так не выйдет, т.к. команды прерывания устанавливают верное значение порта из системной переменной BANK-N. Таким образом, вместо OUT 32765,X надо давать POKE 23388,X. Мой совет Евгению - поменять это в загрузчике. Если не будет успевать проходить прерывание (в чем я очень сомневаюсь, но все же…), поставьте после каждого POKE ещё PAUSE 1.

****

ИФК: На вопрос Кашникова В.Н. из г. Кинель, о выводе сообщений из машинного кода (ZX-РЕВК)-94, № 1, стр. 60) отвечает Вадим Бодров. Он приводит программу "PRINT MESSAGE", которая является практическим ответом на этот вопрос. Ниже представлен её листинг, набранный при помощи ассемблера GENS.

```
1           NOP     ;FN P(X)/PRINT/
2           ORG     63300
3           ENT     $
4           PUSH    AF
5           PUSH    HL
6           PUSH    BC
7           PUSH    DE
8           PUSH    IX
9           LD      A,#02
10          CALL    5633
11          LD      DE,Start
12          LD      IX,(#5C0B)
13          LD      A,(IX+4)
14          LD      C,#00
15          LD      B,A
16 LOOP     LD      A,(DE)
17          INC     DE
18          LD      H,A
19          LD      A,B
20          SUB     C
21          JR      NZ,ADDR_2
22          LD      A,H
23          CP      #80
24          JR      C,ADDR_1
25          RES     7,A
26          RST     #10
27          POP     IX
28          '       POP   DE
29          POP     BC
30          POP     HL
31          POP     AF
32          RET
33 ADDR_1   RST     #10
34          JR      LOOP
35 ADDR_2   LD      A,H
36          CP      #80
37          JR      CLOOP
38          INC     C
39          JR      LOOP
```

В строке 11 в регистровую пару DE засылается адрес расположения набора сообщений.

Сама программа - перемещаема, единственным числовым параметром в ней является ***Start*** - адрес начала таблицы сообщений, которые располагаются одно за другим. Для их разделения у последнего байта каждого сообщения включается 7-й бит, то есть значение байта увеличивается на 128.

Если в строке 25 вместо "RES 7,A" поставить "CPL", тогда маркером конца сообщения должен будет служить "инвертированный" байт: если последний байт текста сообщения имеет код n, то последний символ в сообщении должен иметь код 255-n, то есть быть дополнением его до 255.

Программа загружается в память с любого адреса: LOAD "" CODE adr. Затем выполняется инструкция Бейсика DEF FN P(x), где x - номер сообщения. Процедура извлекает его из области системных переменных и заносит в аккумулятор. Этот способ описан в книге "Элементарная графика". Перед использованием процедуры не забудьте только задать значение ***Start*** - начало таблицы сообщений (строка 11).

ИФК: Мажников Д.В. из города Петровск-Забайкальский интересуется опытом работы с дисками.

КОРР: Хочу обратиться ко всем опытным пользователям TR-DOS. Поделитесь, пожалуйста, опытом, как Вы

используете свои диски. Как дольше они хранятся: отформатированные обычным способом, или "FAST". Следует ли пользоваться очистительными "утками" (игра слов: русское "диск", и английское "duck") и не вредят ли они головкам дисководов? Есть ли способ восстановить диски с ошибкой в 9-ом секторе 0-й дорожки? Проблема кажется тривиальной в большом городе, но если в городе всего три "SPECCY" с "TR-DOS", причём у одного компьютер сгоревший уже год, а второй подходит к нему раз в полгода, тогда гибель одной программы, а тем более всего диска - маленькая трагедия. Ну а если на диске труды последних месяцев… А иметь копию каждого диска невозможно. Порой руки опускаются.

ИФК: Да уж, ощущение далеко не самое приятное, и очень нам знакомое. Ситуация с потерянными программными материалами, пожалуй, случается с каждым программистом, или человеком, работающим на компьютере, вне зависимости от его модели. Что можно посоветовать? Если нет возможности иметь резервную копию каждого диска, то хотя бы используйте разные диски для разных разделов готовящейся программы. Интенсивную работу по разработке или отладке программы начинайте с утра всегда на чистом "рабочем" диске (на который запишите необходимые в работе утилиты), а вечером - можете перебросить то, что уже "доведено до ума" на другой диск для "промежуточного" сохранения. Операцию MOVE на рабочем диске можете делать только после того, как все нужное переброшено с "рабочего" на "промежуточный" диск и никогда не делайте MOVE в процессе работы в течение дня. Помните, что вероятность гибели диска всегда выше при интенсивной работе, так как происходят многократные повторения записи-удаления с перезаписью нулевой дорожки. Используя "рабочий" диск, Вы, в крайнем случае, потеряете только результаты дневной работы, но не более того. Полностью заполненные диски заново не форматируйте, "наводя порядок" и заранее подготавливая для работы в будущем. Возможно, Вам ещё придется восстанавливать с них информацию. Кроме того, можно делать резервные копии не самих дисков, а каталогов на них. Уже давно существует программа "CAT", и другие, ей подобные, которые могут сохранять нулевую дорожку на 79 треке или за 80 треком или в файле на другом диске, и восстанавливать её при необходимости. Ведь в 95% случаев портится именно системный нулевой трек, а не какой-то конкретный файл. Чем чаще Вы сохраняете текущий нулевой трек, тем больше шансов сохранить программы. Таким образом, можно обойти и ошибку в девятом секторе.

Способ форматирования "FAST" отличается от обычного только расположением секторов на дорожке (подробнее об этом см. ZX-РЕВЮ № 4, 1994.) Так что, никакого дополнительного износа здесь быть не может. Напротив, время работы дисковода сокращается.

Очистительные дискеты, как и подобные кассеты, работают благодаря эффекту трения. И, разумеется, частое их использование приводит к сокращению срока службы дисководов. Эффект же от их использования не слишком велик, он сопоставим с результатом работы очистительных магнитофонных кассет - вручную можно очистить гораздо лучше. Но, так как лезть в дисковод для очистки не всегда удобно, то такими дисками можно пользоваться в меру.

Мажников Д.В. продолжает свое письмо.

КОРР: Обращаюсь к Вам, ИНФОРКОМ, ещё с одной проблемой. Кое-как я раскидал на диски все игрушки по тематике, взяв за основу и несколько упростив систему классификации от "SINCLAIR USER", о которой Вы писали в 1991 году. Сделал я так: помня о том, что в заголовке диска всего 8 байтов, я "обозвал" их так: T-NN(XX), т.е. T - это "TOYS", NN - номер диска (HEX), XX - одно из сокращений:

AA - аркадные адвентюры (T.BIRDS 1-4, DIZZY 1-6, TERRA- MEX);
AR - аркадные программы (PACMANIA, MAD MIX 2);
AC - action (REX 1,2; EXOLON);
SI - simulators (F-19, TOMAHAWK);
PU - puzzle (PUZZNIC, KLAX, STACK UP);
TR - traditional (PSI-CHESS, 100&80);
ST - strategy (ZULU WAR, SIM CITY);

Упрощение коснулось лишь оценки игр. Теперь голова болит, как рассортировать системные программы. Куда отнести различные boot 'ики? А программу "L.B.P." Бессонова? Что входит в понятие "прикладные программы"? Не могли бы Вы хотя бы вкратце рассказать, как можно рассортировать системные программы.

ИФК: Сохраняя стиль Вашей классификации, здесь применим такой же метод разделения программ по назначению:

UT (UTILITY) - утилиты (программы типа DISK-DOCTOR, DCU, ADS и др.);
SH (SHELL) - оболочки (те же boot'ы);
TX (TEXT) - программы для работы с текстами;
GR (GRAPHIC) - программы, работающие с изображением;
MU (MUSIC) - музыкальные программы и т.д.

Возможно, у кого-то из читателей уже сложилась своя, четкая система классификации и он готов ей поделиться?

Мажников Д.В. в своем письме поднимает также вопрос отечественного рынка программного обеспечения.

КОРР: Очень печально, что из Европы больше практически не поступает свежих игрушек. Нам остается, надеется на себя (что, по-моему, бесполезно, или положение исправится очень и очень не скоро. У нас нет того, десятилетнего опыта программирования, что есть у них; а 30-40 наших и оригинальных игрушек положение не спасут).

ИФК: На наш взгляд, положение не столь печально. В последнее время на "Спектруме"' появились новые высококлассные программисты, создающие свои игры для этого компьютера. Примером может служить хотя бы

проект "STAR INHERITANCE" (см. "РЕВЮ № 3,1994), который активно готовится сейчас к выпуску под эгидой редакции журнала "SPECTROFON". И это далеко не единственный из известных нам новых проектов.

КОРР: В то же время, "ZX-SPECTRUM 48" служит лишь для разгона, большинство хотят иметь "128К + TR-DOS + AY…". Наверняка, в ближайшие годы "SP-128" станет необходимым минимумом. Сейчас сложилась такая ситуаций, когда ряд великолепных старых игр уже "не тянут" классом на 1994 г. И если по графике мы мало что можем улучшить, то по звуку - целый океан возможностей. Например, я заново открыл для себя "A.T.F.", "EXOLON" и много других игр, послушав их работу. Я вот к чему веду: Было бы здорово начать работу в этом направлении, и делиться на страницах "ZX-РЕВЮ" открытиями, доработками, находками. Но, а если кто-то уже "озвучил" какую-то игру, он может продать её с Вашей помощью. Думаю, многие приобрели бы "TOMAHAWK-128" и т.п. Яркий пример достижимости этого - новые версии "ELITE". А может, кто-то обладает достаточными знаниями об AY, и его программировании из машинных кодов, чтобы поделиться ими со страниц "ZX-РЕВЮ"…

И ещё одно предложение владельцам SP-128. Господа! Продать можно не только игру, написанную или переделанную Вами, но и ЕЁ КУСОЧКИ! Можно предложить к продаже хорошую музыку для AY, написанную Вами, или Вашим талантливым другом-композитором. (Например, игра "PRESIDENT'S GUARD", предложенная в ZX-РЕВЮ № 1, 1994. Тематика игры мне чужда (да простит меня автор), но музыка, написанная музыкантом - это нечто стоящее!).

Продать можно хорошую заставку (мне до сих пор хочется взглянуть на картинки "CLOTH", "ROBOT", "SUNRISE", речь о которых шла в РЕВЮ № 9'91 стр. 198, И которые так заинтересовали "ИНФОРКОМ"). Продать можно даже спрайты (Это не шутка! Те, кто пытался изобразить что-нибудь типа героев или предметов из игр подобных "DIZZY" или "NIPPER" поймут меня). И пусть прибыль от этого невелика, но все же лучше, чем, если всё это будет пылиться у Вас на дисках без дела.

Более того, можно продать хороший сценарий игры! Опишите в нескольких словах свою идею, и не опасайтесь, что её "слижут", ведь тот, кто не может написать хороший сценарий, не сможет детализировать даже готовую идею.

Одним словом, я предлагаю, не в ущерб рынку программ, создать рынок идей и талантов! Это сослужит всем нам хорошую службу, когда Российские фирмы по производству программного обеспечения, наконец, встанут на ноги, и начнут искать таланты.

ИФК: Наш корреспондент прикоснулся к действительно важной и интересной теме. Конечно же, потребность в рынке составных частей программ давно назрела. Обладая доступом к большой библиотеке программных модулей, программист может уже не отвлекаться на создание музыки или прорисовку спрайтов и сможет спокойно сконцентрироваться на создании самой программы. И что особенно приятно, многое из описанного нашим читателем уже начинает создаваться.

В-первых, сейчас появилось большое количество фирменных программ, расширенных нашими программистами до режима 128 килобайт. Тут и "LEE ENFIELD 128", и "TRANTOR 128", можно назвать ещё с десяток только последних поступлений. Музыку вставили даже в "ERIC AND THE FLOATERS"! Хочется только посоветовать молодым программистам не увлекаться озвучиванием игр с помощью всем известных, избитых мелодий, а искать что-нибудь новое (например, на описанном выше рынке модулей).

Более того, в последнее время появились фирмы, заинтересованные в покупке таких продуктов, как графика, музыка и программная поддержка. Так что, путь открыт, осталось его расширить и сделать более цивилизованным, а мы со своей стороны постараемся оказать этому всестороннюю поддержку со страниц нашего журнала.

А каково Ваше мнение, уважаемые читатели?

****

ИФК: Шалыминов С.П. из Нижнего Новгорода делится советами по игре ACADEMY.

КОРР: Автоматические заводы я не взрываю так, как это описано в РЕВЮ-94 № 3. Это рискованно, могут убить. Я делаю иначе: сначала уничтожаю весь "эскорт" заводов, чтобы не мешали. Потом не очень быстро подлетаю к крайнему заводу так, чтобы не убили остальные. Иду до тех пор, пока завод не оказывается на сканере. Далее, разворачиваюсь на 180 градусов, бросаю бомбу и быстро ухожу. Причем, за всё время манёвров я стреляю из лазера через 1-2 секунды - и, естественно, сразу, как только летит ракета с завода. Причем, заводы выпускают ракеты только в одну сторону. Подходя к заводу, я замечаю это направление, и иду по касательной в этом направлении к заводу. Таким образом, я вижу момент выхода ракеты, и могу уничтожить её лазером. Но самое любопытное, что ракета всегда норовит попасть в передний экран (или в задний, если Вы развернуты). Поэтому не нужно паниковать - лазер её уничтожит почти в упор.

После разворота нужно стрелять осторожно, чтобы не попасть в свою же бомбу - это ее уничтожит.

В настоящий момент я на третьем уровне игры, но не собираюсь останавливаться.

ИФК: Шалыминов С.П. спрашивает, может ли кто-нибудь помочь с приобретением "ELITE 2 (3,4)" из Новосибирска?

КОРР: Я хотел бы поиграть в неё, но не могу никак достать.

Да, по поводу "ELITE". В моей версии "ELITE III (THD)" очень много астероидов с отшельниками. Иногда их можно отличить от обычных астероидов, но не всегда. Я обычно делаю так: выстрелю один раз по нему коротко и смотрю - если он начнет маневрировать, или, тем болев, стрелять, - значит это "STRANGER". Я обычно в таких случаях набираю скорость и лечу к планете. Как правило, при появлении пиратов он сразу же улетает. Если же по нему слишком "сильно" выстрелить, то он начинает выпускать корабли, причем, пропорционально выстрелам. Был случай, когда один отшельник выпустил сразу же 6 кораблей - еле отбился.

ИФК: Существует вполне точный способ распознать отшельников. Если система на Вашем корабле заглушила J-двигатели перед астероидом, и он находится близко от Вас, приблизительно в центре экрана - значит это обычный астероид. Если же он далеко, да ещё и смещён на локаторе в сторону, так, что его не видно на экране без разворота - значит это населенное небесное тело со своим источником энергии, двигателями и малыми кораблями. Кстати, шесть кораблей на астероиде - это далеко не предел. Количество атакующих кораблей возрастает с ростом Вашего боевого рейтинга.

КОРР: В этой версии тоже бывает взаимодействие кораблей. Как-то раз я выстрелил по астероиду, а он ответил. Ну и ладно. Я молча продолжал лететь. STRANGER долго вился вокруг меня, иногда постреливая. Но впереди появился другой астероид. Я решил не рисковать, и не обратил на него внимания. Вдруг, на моих глазах, он резко взмыл "вверх" (относительно моего корабля). Я быстро переключился на боковой экран, повернулся на 90 градусов, и увидел интересную сцену: эти два отшельника летали рядом друг с другом, как будто были на одном стержне, и этот стержень крутился вокруг своей середины (переговоры!) По том они разлетелись в разные стороны, и стали меня "обрабатывать" уже вдвоём, причём летали они по сложным траекториям, но каждый раз я не мог их поймать в один экран. Вот такая детективная история.

А однажды я увидел пульсирующую станцию! Около неё не было кораблей, а когда я летел к планете, от неё, как на пожаре, летели разные корабли, я даже J-двигатели не смог ни разу включить. Когда я подлетел к станции (я не стал пользоваться DOCKING COMPUTERS, т.к. меня заинтересовало, почему корабли так бегут от планеты) она ритмично, около двух раз в секунду, изменяла объём: нормально - меньше - нормально - больше - нормально. Причём, изменения относительно нормального были около 1/2 объёма.

Я попытался влететь - но, увы, меня убили при входе - может быть, я задел за стенки тоннеля. Но когда я снова полетел туда та же картина; я воспользовался стыковочным компьютером, но все равно погиб. Может, кто знает, что это было?

ИФК: С таким эффектом мы не сталкивались. Для наших читателей было бы интересно узнать, какое сообщение Вам выдано при использовании стыковочного компьютера, в какой звёздной системе это происходило, в какой галактике, а также Ваш рейтинг. Если Вы сообщите нам это, то, возможно, кому-то из читателей удастся повторить этот эффект и пролить свет на новую загадку вселенной.

Далее, я прошел игру "GAME OVER", она выдала код: 18024. Это для "GAME OVER 2"? Кто-нибудь её видел?

ИФК: Да, такая игра существует, это продолжение первой части.

****

ИФК: Несколько слов об **ELITE-3** (версии для 128К московской группы "**THD**"), упоминавшейся в ФОРУМЕ, хочет сказать Михаил Кретов из г. Златоуста Челябинской обл.

КОРР: Кроме тех особенностей, которые были описаны в статье, отмечу следующее. Если Вы, выполнив миссию по спасению беженцев, отгрузите состояние на диск, то будете обречены, выполнять первую миссию каждый раз при перезапуске игры - после загрузки или после гибели. Кстати, если Вы, имея один или несколько SUPPLEMENTARY CARGO BAY, согласитесь спасти беженцев, то после избавления от REFUGEES обнаружите, что грузоподъемность корабля уменьшилась до стандартных 35 тонн. Если же у Вас хватит терпения, и Вы доберетесь до выполнения 2-й миссии, а затем опять выгрузитесь на диск, то после перезапуска игры в списке вооружений не будет ничего похожего на CLOAKING DEVICE _"шапка-невидимка" просто потеряется, и Вам снова предложат выполнить (после выполнения или отказа от 1-й миссии) вторую миссию, и так до тех пор, пока Вам не надоест эта свистопляска и Вы не пересядете на менее интересную, но, по крайней мере, нормально работающую версию ELITE. Описанные выше "странности" этой версии игры лично я не могу объяснить ничем другим, кроме как недостаточной продуманностью изменений, внесенных группой "THD" в старую добрую ELITE. Но, несмотря на высказанные выше претензии, хочу сказать, что играть в эту версию игры гораздо интереснее, чем в "обычные" версии (до тех пор, пока не возникнут проблемы с выгрузкой).

ИФК: Можно было бы предположить , что указанное явление является каким-то сбоем программы или следствием несовместимости, но точно такие же претензии к этой версии ELITE содержатся в письме Романа Озеранского из Москвы. Видимо, это недоработка программы.

****

ИФК: О "жучке" в игре HERO QUEST нам сообщили практически одновременно два читателя: Евгений Черемных из Красноярска и Андрей Сюткин (возможна ошибка - фамилия неразборчива) из Кургана. Они приводят способ, позволяющий обессмертить персонажей игры.

КОРР: Для этого надо сделать следующее.

1. Погубить персонаж (сделать так, чтобы у него стало 0 BODY POINTS) - появится сообщение, что он погиб.
2. Войти в меню CREAT CHARACTER (погибший персонаж выделен надписью DEAD вместо имени) и переименовать героя, для этого надо навести указатель на DEAD и набрать имя (любое). Ваш персонаж после этой процедуры оказывается жив, есть деньги и амуниция.
3. Загрузите любой уровень и, не применяя никаких средств для увеличения силы, станьте персонажем на необнаруженную ловушку (TRAP) - компьютер сообщит Вам, что Вы попали в ловушку, при этом Вы потеряете 1 единицу энергии и у Вас станет вместо нуля - 255 единиц энергии (вместо надписи о смерти Вы увидите в квадрате энергии всякую ерунду, но не обращайте на это внимания).

После этого персонаж становится псевдо-бессмертным (получает 255 единиц энергии), но ни в коем случае

нельзя применять средства для увеличения силы.

Этот фокус Вы можете сделать сразу для трех персонажей, так как одного (лучше Barbariana) нужно оставить со всеми силами для того, чтобы он защищал других от ловушки.

ИФК: Кстати, Евгений Черемных задает вопрос: является ли HERO QUEST - RPG? Да, конечно, хотя и простейшей.

Между тем, Евгений продолжает делиться своим опытом.

КОРР: POBOCOP - на 1 и 3 уровнях необходимо убить бандита, прикрывающегося заложником (на 1 уровне женщиной). Если Вы смогли подстрелить бандита на 1-м уровне, Вам не составит труда сделать это и на 3-м, но необходимо стрелять и быстрее и точнее. Я пользуюсь бессмертной версией и, поэтому, время от времени не могу стрелять (как будто кончаются патроны), но это не мешает после двух-трех попыток уничтожить бандита.

SIM CITY - отгрузочный файл состоит из двух частей: заголовка (длина 31 байтов) и блока 9216 байтов. Заголовок:

1-12 байты - имя (набранное при выгрузке состояния);

13-16 байты - количество денег (например, #FF #FF #10 #10 равно 1 311 512 $);

17-20 байты - год;

21-31 байты - название города. Заголовок одинаков для магнитофонной и для дисковой версий. Никакого контроля (подсчета контрольной суммы) программа не производит.

В блоке находится карта, но тут пока что ещё ничего не известно.

ELITE - в версии Ml28 возможен гиперпереход через станцию. В отличие от версии Родионова, где надо вылетать со станции, поворот на 180°, полный газ и нажать клавишу "H", - в версии M128 все по-другому. Нужно вылететь со станции, развернуться на 180° и потихоньку начать стыковку. Когда стыковочное отверстие займет почти весь экран (это происходит перед самым входом в станцию), нажмите "H" - и Вы на другой станции.

ИФК: А Евгений и Алексей Федяевы из Москвы сообщают, что в ELITE (версии 3 по классификации ZX-РЕВЮ-92, стр. 84), с помощью имитации отгрузки (как это было описано в ZX-РЕВЮ-94, № 5, стр. 58), можно не только избежать захвата корабля пиратами, но и выйти из других "безвыходных" ситуаций: при гибели на подлёте к планете или в бою с таргонами после выхода из гипертуннеля.

****

ИФК: В ответ на вопрос в ZX-РЕВЮ-93 № 11-12 читателя Федора Юхновича о программе ENTERPRI (ENTERPRISE), Мизинов Антон готов поделиться тем, что знает.

КОРР: В начале игры, сразу после загрузки, Вас просят ввести имя пилота, это надо сделать обязательно, но об этом позже. После выбора джойстика или клавиатуры Вы оказываетесь около вращающейся планеты (имею ввиду ваш корабль). Кроме прямого управления кораблем есть ещё клавиши управления разными приборами:

M - карта расположения планет, если на планету навести курсор и нажать "огонь", то получите информацию об этой планете (её координаты по X,Y,Z, дистанция до планеты, честность в торговле, полезность в торговле, минеральное основание).

B - включение/выключение двигателей межпланетной тяги.

R - информация о физическом состоянии приборов и обшивке корабля.

OK - нормальное;

NOT OPERATIONAL - не работает;

VERY BADLY DAMAGET - полностью разрушено.

E - открытие/закрытие люка грузового отделения.

Z - вкл/выкл энергетической защиты.

I - информация о Ваших товарах и наличных.

C - бортовой компьютер.

H - пауза.

J,K - смещение порядков координат.

ВВЕРХ + ОГОНЬ - движение вперед (скорость).

ВНИЗ + ОГОНЬ - движение назад. В СТОРОНУ + ОГОНЬ - вращение вдоль продольной оси.

Если выдается сообщение "STOP! POLICE!", то лучше остановиться, иначе с Вас возьмут штраф. Перед входом в атмосферу планеты выключите межпланетные двигатели. После входа в атмосферу космический пейзаж сменится горным ландшафтом и впереди будет видно точку - это взлётно-посадочная полоса (ВПП). Смело летите к ней. Когда приблизитесь, уменьшайте ускорение до 10 и приземляйтесь, как в других имитаторах полётов. Снижайте ускорение до 0 и ждите, пока не остановитесь.

Когда остановитесь, открывайте люк грузового отделения. На ВПП поймут, что Вам что-то нужно и пришлют Вам доброжелательное существо для беседы. Он, Она или Оно будет вести с Вами диалог (лицо инопланетянина будет видно на месте радара переднего обзора), его мимика и звуки, издаваемые компьютером дополнят ощущения присутствия иного разума. Автоматически они будет переводиться и высвечиваться на английском языке. Не смотря на его доброжелательность, лучше не пробовать ему грубить - сначала он их проигнорирует, но потом можно услышать и предупреждения… Предложения можно вводить самому, а можно воспользоваться готовыми:

S.SHIFT+Q - HOW MUCH FOOD

S.SHIFT+W - DO YOU WANT ANY… (Вам надо…?)

S.SHIFT+R - I HAVE… (Мне надо…)

S.SHIFT+T - DO YOU HAVE ANY… (У Вас есть…?)

S.SHIFT+Y - HOW MUCH DO I OWE YOU (Сколько я Вам должен?)
S.SHIFT+A - HOW MUCH IS YOUR FOOD? (Сколько стоит еда?)
S.SHIFT+S - HOW MUCH IS YOUR FUEL? (Сколько стоит топливо?)
S.SHIFT+D - HOW MUCH IS YOUR INSURANCE (Каковы расходы по страховке?)
S.SHIFT+F - HOW MUCH FOR REPAIR? (Сколько стоит ремонт?)
S.SHIFT+G - HOW MUCH WILL YOU GIVE FOR. (Сколько Вы дадите на…)

Когда Вы начинаете говорить, инопланетянин, в основном, либо говорит, что будет помогать Вам первое время, либо спрашивает о том, как Вы долетели. В переговорах ведите речь только о торговле или ремонте, иначе Ваш разговор превратится в глупый обмен фразами. Тут (в торговле) и нужно имя, вводимое в самом начале: при покупке чего-либо Ваш собеседник спросит Ваше имя, если его назвать неверно, то увидите такую фразу: "THIEF! REPAY ME OR I'LL CALL THE POLICE" (Вор! Расплатись, иначе я вызову полицию!), если же Вы введете верно, то он сообщит, что иметь дело с Вами - одно удовольствие.

На сообщение клавиши D он выговаривает таблицу:

FOOD ONLY COVER	1	(расходы только на еду)
FUEL ONLY COVER	9	(расходы только на топливо)
FOOD AND FUEL COVER	8	(расходы на еду и топливо)
SHIP ONLY COVER	24	(расходы на корабль)
SHIP AND FOOD COVER	21	(расходы на корабль и еду)
SHIP AND FUEL COVER	28	(расходы на корабль и топливо)
SHIP, FOOD, FUEL COVER	27	(расходы на корабль, еду, топливо).

На клавишу F инопланетянин говорит цены на запчасти и спрашивает, что Вам конкретно надо. Следует нажать клавишу с номером запчасти. Но надо следить за своими расходами. В каких запчастях Вы нуждаетесь, можно узнать после посадки клавишей R. Если Вы хотите вылететь с этой планеты, то когда настанет Ваша очередь вводить реплику, нажмите BREAK, перед Вами снова откроется горный ландшафт, набирайте скорость и взлетайте с ВПП. Угол отклонения от земли ставьте не более 45 градусов. В открытом космосе на пути к другим планетам можно встретить: корабли полиции (приплюснутый тетраэдр с очертаниями ангара на плоской поверхности для стыковки). Они обычно останавливают для проверки и отпускают. Ещё на пути можно увидеть кувыркающиеся группы астероидов по 7 штук. Планеты состоят из точек, расположенных как бы на перекрестье параллелей и меридианов, поэтому производят впечатление шара.

Это всё, с чем я разобрался, но для меня до сих пор остается загадкой, чем там надо торговать, что ищут полицейские, можно ли там совершать гиперпрыжки - искать планеты по координатам очень сложно. Но это вопросы к экспертам, а также я хотел бы узнать что-нибудь о программе DRILLER, об "ELITE", но на IBM-совместимых компьютерах.

****

ИФК: Зинков Сергей из г. Абакан отвечает на вопрос Александра Гмарь по поводу игры "TITANIC-2".

КОРР: В комнате, где водолаз всё время погибает, нечего делать. Это холодильник, и его авторы придумали специально, чтобы сбить игрока с толку. Сначала надо включить рубильник внизу, к которому можно попасть через пробоину в обшивке корабля. Рубильник откроет одну из дверей в корабле, которая закрывала доступ к другой пробоине, и взрывчатке в другом отсеке внизу. Когда Вы возьмёте взрывчатку, откроется дверь в каюте капитана. Если Вы подплывете с взрывчаткой к сейфу, произойдет взрыв и документы - Ваши.

****

ИФК: Локтюхин Максим из г. Саратов- 8 готов помочь всем, у кого возникли трудности с прохождением игры "DRACONUS". Вот его адрес:

41008, г. Саратов-8, ул. Политехническая 31/41, кв.69,
Локтюхину Максиму

Он также написал, что для превращения дракончика в рыбу, необходимо взять кристалл, а потом, встав на площадку над водой, нажать одновременно "вперед" и "вниз".

****

SNOOPY (первый уровень)

ИФК: Советы по этой игре принадлежат Дмитрию Иванишеву.

1. Взять миску (BOWL) и съесть всё из неё.
2. Взять в своей конуре печатающую машинку (TYPEWRITER) и на печатать на ней письмо.
3. Взять письмо (LETTER) от дать его пацану, который стоит в соседней комнате, он бросит его в почтовый ящик.
4. Взять банку с вареньем (CO OKIE JAR), съев варенье, идти к реке.
5. У реки поймайте в банку лягушку.
6. Этой пойманной лягушкой нужно напугать девочку, она уронит резиновую уточку! (Но Вы её не берите).
7. Возьмите мяч (BALL) и бросьте его в реку в том месте, где не можете перепрыгнуть. Затем возвращайтесь за рогаткой.
8. Взяв рогатку (CATAPULT), идите через речку к колодцу. Там, сбив шарик, Вы можете взять насос.
9. Имея насос (PUMP) и резиновую уточку (RUBBER RING) - вы можете накачать её под деревом, чтобы взять змея, и возле тучи с дождем, чтобы взять кирпич.
10. Взяв кирпич (BRICK) вы мо жете сбить яблоко (APPLE) с дерева (где это яблоко применять - не

известно).

11. Взяв воздушного змея (KI TE) отдайте его пацану, который относил письмо, он даст Вам ключ.

12. Ключом (KEY) откройте в школе шкафчик, из него вывалится одеяло.

13. Одеяло (BLANKET) - отдать пацану, сидевшему в ванной (который его просил). После этого компьютер встает на загрузку с магнитофона: "START THE TAPE" - первый уровень игры пройден, а второго у меня нет.

Есть в игре BUST (бюст), но не понятно, где его применять.

Есть часы, которые показывают время. Отсчет у них начинается с 10:01:00 Am.

****

LODE RUNNER

ИФК: Совет по этой игре прислал Сергей Колесников из Ставрополя.

КОРР: Очень многие не удовлетворяются заданными экранами и создают свои (благо, такая возможность есть). Кстати, в оригинале LODE RUNNER 2 есть ошибка - 64-й экран вообще не проходится. Так вот. Загрузка своих измененных или вновь созданных экранов с ленты занимает дополнительно около трёх минут. Если Вы хотите избежать этого, приделайте к отгруженному блоку экранов заголовок (для его записи подайте команду: SAVE "PICTURES" CODE 44836,20700). Теперь очистите память компьютера и наберите программку:

```
10 CLEAR 24575: LOAD "lode 2" CODE
20 LOAD "PICTURES" CODE
30 SAVE "lode 2" CODE 24576, 40959
```

Новый блок кодов "lode 2" CODE содержит сгенерированные Вами экраны и времени на их догрузку не требуется.

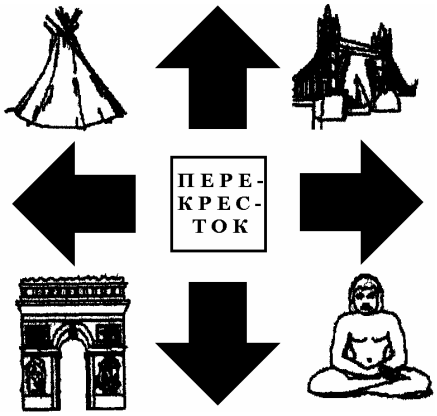
****

ИФК: Александр Косых из Красноярска недоумевает по поводу игры PREDATOR 2.

КОРР: На последнем уровне автор подложил нам хорошую "свинью" - он сделал PREDATORa бессмертным. Хочу спросить - может быть, кто-нибудь знает, как пройти этот уровень?

****

ИФК: Д.Губанов из г. Минусинска (Красноярский край) просит помощи читателей по игре THE HUNT OF THE RED OCTOBER - он пишет, что будет рад любой информации.



Итак, скитальцы по затерянным компьютерным мирам снова собрались на распутье, под раскидистым деревом, чтобы поделиться своим опытом и обменяться полезными советами. Прислушаемся и мы к их содержательной беседе.

THE HOBBIT.

Тахир Габидулин из г. Глазова отвечает на вопрос нашего читателя в одном из прошлых выпусков "РЕВЮ" о том, как переплыть Чёрную реку?

Переправляться через реку надо так: посмотрите поперёк реки - нет ли там лодки? (LOOK ACROSS RIVER). Если есть, то бросайте к ней веревку (THROW ROPE ACROSS RIVER), проверьте, попала ли веревка в лодку. Если да, то тяните её (PULL ROPE), и лодка - Ваша.

А на вопрос И.А. Киселева из Самары отвечает Кунгуров А.Н. из п. Нижний Архыз.

Влезать в лодку и вылезать из неё можно при помощи фразы CLIMB INTO (OUT) BOAT.

Алексей Хамардюк спрашивает, каким образом можно убить дракона? Лук и стрелы он взял, но места, свободного от чешуек, как в книге, найти не удалось.

Павел Бочарников из г. Южно-Сахалинск тоже прочно "засел" в этой игре и взывает о помощи. Проблемы у него такие:

Как пройти место с мощными подъёмными воротами (PORTCULLIS) в замке эльфов, которые придавливают хоббита сразу же, как только он появляется в этой локации. Попасть же сюда можно двумя способами: переплыв реку или спустившись сквозь открытый люк в винном хранилище. Здесь же можно встретить MAGIC DOOR, которая непонятно чем открывается.

Кроме того, интересна локация "THE MOUNTAINS". Если здесь пройти на восток, то компьютер сообщит, что проход завален.

И ещё, Павел хотел бы найти друзей по "Спектруму", которые живут в Южно-Сахалинске (чтобы не только переписываться, но и меняться программами):

тел. в Южно-Сахалинске: 96-8-91

Мажников Д.В. просит начать атаку на игры ZOMBI и MUNSTERS.

В игре ZOMBI несколько человек должны на вертолете покинуть здание супермаркета, окруженное зомби. В игре MUNSTERS девушка должна очистить дом от нечисти. Игры очень сложные и интересные.

Игра **ZOMBI** была создана по кинофильму "Рассвет мертвецов". Суть его в следующем. Неожиданно на всём земном шаре происходит катаклизм - все недавно умершие люди оживают после смерти, и превращаются в зомби. Зомби охвачены только одним желанием - жаждой убивать людей. Укус зомби смертелен для живого человека, и тот, в свою очередь, сам превращается в живого мертвеца. Героям игры - троим мужчинам и девушке удалось спастись от зомби на вертолёте. Когда в вертолёте подошло к концу горючее, он приземлился на крышу супермаркета. Отсюда и начинается игра. Вам необходимо заблокировать входы в магазин и истребить всех зомби (лучше бить в голову). Чтобы они снова не воскресали, их надо поместить в морозильник в подвале. Вам придется столкнуться также и с бандой рокеров, которые воспользовались царящей повсюду неразберихой и грабят население. Игра действительно очень интересная, а в сборнике "ЛУЧШИЕ ИГРЫ ZXSPECTRUM", выпуск 2-3, дано её описание.

В игре **MUNSTERS** главный герой - Франкенштейн, именно его Вы должны сначала воскресить при помощи девушки, а потом сопровождать его при помощи других персонажей. Описание дебюта к этой игре (прохождение первого уровня) опубликовано в электронном журнале "SPECTROFON № 6, 1994".

Обе игры принесут Вам достаточно сюрпризов и неожиданностей и мы обязательно опубликуем все письма читателей с новой информацией по ним.

А.В. Журба из г. Екатеринбурга просит разобраться на страницах журнала с такими играми, как LORD OF THE RINGS 1,2, SHADOWS OF MORDOR, HOBBIT. И ещё: Что нужно делать в игре HIJACK? Как убедить Лестрейда, что Базиль имеет военные планы? Что нужно сделать в играх MESSAGE FROM ANDROMEDA и GRANGE HILL?

Кроме того, наш корреспондент предлагает свою помощь в совместной раскрутке адвентюрных игр. Вот его адрес:

620086 г. Екатеринбург ул. Пальмиро Тольятти, д.15-в, 52, Журба А.В.

Павел Ловыгин из г. Костромы делится своими успехами в некоторых адвентюрах.

Игра **HE-MAN**.

1. Сначала вы должны найти меч силы (SWORD OF POWER). Для этого Вам надо зайти в спальню и исследовать кровать. После этого Вы увидите, что она состоит из столбов (POSTS). Исследуя столбы, Вы откроете проход. Зайдя в него, Вы окажетесь в маленькой комнате, и найдете там тайник (RECESS). Исследовав его, Вы найдете меч. Теперь осталось лишь взять его.

2. Теперь Вам надо попасть в замок Серого Черепа. Подойдя к стенам замка. Вы попадете ко рву с водой. Теперь переплывайте ров (SWIM), и Вы окажетесь у входа в замок. Там Вы увидите останки мертвых солдат: черепа

(SCULLS) и одежду (COATS). Чтобы попасть в замок, нужно ударить по одежде (HIT COATS) и подъёмный мост опустится. Теперь Вы можете смело идти в замок (GO TO JAWBRIDGE).

И несколько полезных советов. Исследуя людей (EXAMINE PEOPLE) в гостинице (MAN OF ARMY), Вы найдете знакомого солдата. Поговорите с ним (TALK TO MAN). Он откажется с Вами идти, но даст Вам оружие (MOLECULATOR). Но я так и не смог им воспользоваться.

Также можно пройти сквозь зеркала (GO MIRROR).

EVIL-LYNN надо убивать так: Зайдя в экран, она сразу же заколдовывает Ваши ноги. Вы должны просто подпрыгнуть (JUMP), и тем самым разорвать её чары (она погибнет).

Многих чудовищ можно убивать с помощью меча (HACK … WITH SWORD). Чтобы убить всех скорпионов, надо два раза дать команду "HACK SCORPIONS". TIME-LINK можно найти в саду (EXAMINE GARDEN) и за зеркалами, где находится чудовище (MANTENNA). Сначала убейте его мечом, а затем исследуйте сталактит (EXAMINE STALACTITE). Часы в замке можно открыть (OPEN CLOCK). Чтобы Вас пропустили TWO-BAD, надо дать им бирюзу (TURQUOISE).

Игра **GREMLINS**. На кухне, использовав пульт управления (REMOTE CONTROL), Вы найдете в ящике (DRAWER) нож и зажигалку. Теперь Вам надо вернуться в спальню и убить гремлина, бросающего в Вас дротики, с помощью ножа. Теперь возьмите фонарик (FLASHLIGHT) и идите на бензоколонку (PETROL STATION). Там возьмите баллон с газом. Идите к почтовому ящику. Теперь делайте так:

1. Зажгите фонарик (LIGHT FLASHLIGHT) и засуньте его в ящик (INSERT FLASHLIGHT). Оттуда сразу же выскочит и убежит гремлин. Если Вы этого не сделаете, гремлин будет мешать Вам разбирать ящик.

2. Откройте клапан на баллоне с газом (OPEN VALVE) и зажгите автоген с помощью зажигалки (BURN TORCH).

3. Теперь режьте ящик (CUT BOX). Вы разрежете его на металлические пластины (METAL PLATES).

Дальше Павел разобрался лишь частично:

Фотоаппаратом (CAMERA) можно отпугивать гремлинов. Нажимая кнопку (PRESS BUTTON), Вы приводите в действие вспышку на фотоаппарате.

Пилой можно распилить трубку в баре (PIPE), а также на баллоне с газом (CUT PIPE).

Чтобы начался фильм, надо включить проектор (START PROJECTOR).

Лестницу использовать в универмаге (STORE) на самом верхнем этаже. Там будет люк. Поднимайтесь (GO TO TRAPDOOR) и Вы окажетесь на крыше (ROOF). Лестница сама установится, и будет стоять на месте, пока Вы её снова не возьмете. Гизмо (GIZMO) уползает в вентиляционное отверстие (VENT) там, где оно есть. Оказывается он на верхнем этаже. Он открывает Вам закрытую дверь.

Чтобы спустить воду в бассейне (POOL), надо просто взять пробку и вода сольется.

Если кто-нибудь разобрался больше или хочет сотрудничать, он может писать по адресу:

156016, г. Кострома,
Давыдовский 2, д. 73, кв.28,
Ловыгину Павлу

Просто крик о помощи услышали мы, распечатав письмо Степанишева Е.В. из Лениногорска. "Люди!" - пишет он, - "Знает ли кто-нибудь, как убрать тощего медведя с выступа в THRONE-ROOM? Все это находится в игре **ADVENTURELAMD** из серии **QUESTPROBE ADVENTURE**." Попытки ввести команды "PUSH", "SLIDE", "SHIFT" не увенчались успехом, не помогают также "KICK" и "HIT". Учитывая то, что медведь тощий, я накормил его королевским мёдом (DROP HONEY) и взял зеркало из его лап. Но зеркало, будучи брошенным, на ковер, вежливо намекает, что мёд все-таки тратить не следовало. Тем более что мёд - это одно из сокровищ. Разозлившись на медведя, я бросил в него топором (THROW AXE AT BEAR), но этот самый BEAR всегда уворачивается и разбивает зеркало.

Непонятно, что делать в игре **MOVIE** (фирма "IMAGINE").

Из долгого разговора с людьми и попугаями я понял:

1. Мне все советуют поговорить со священником.

2. Нужно попробовать SECOND DOOR. Но где она и есть ли FIRST DOOR, никто не говорит (наверное, государственная тайна).

3. Говорят также, что для то го, чтобы попробовать вторую дверь (TRY SECOND DOOR) нужно иметь ещё и … (*название неразборчиво - ред.*)

4. На вопрос, "где его взять?", многие отвечают, что это часть головоломки.

5. Есть один несговорчивый товарищ, который требует деньги за оплату сведений. Но когда я разыскал эти деньги и принес их ему, то он не стал ничего брать, не понимая, о чем идёт речь.

Максим Бугрий из г.Старый Оскол добился успеха в программе SPIDERMAN. В этой игре главный герой, человек-паук, путешествует по зданию. На него никто не нападает, но у него нет и друзей. Почти все, к кому он не обратится, говорят: "YOU NEVER GET MY GEM". Только мадам Веб говорит, что может проанализировать (SCAN) кого-нибудь по просьбе героя, а RINGMASTER утверждает, что победить его очень просто, достаточно нажать на выступ и повернуть его. Но, сделав это. Вы слышите лишь звук "CLICK". Такой же звук издает кнопка неработающего лифта.

Успехи:

1. Максим превратил Лизарда в спящего доктора Коинера, бросив в него вещество, которое получил, смешав два других, найденных в маленьком офисе на 3-ем этаже. (Он сделал это в химической лаборатории командой "MIX").

2. В компьютерной лаборатории запустил компьютер (START COMP) - получил клавиатуру, но что с ней

делать, непонятно (она не берётся).

3. Найден один *GEM*. Максим перемещается по этажам с помощью пустой шахты лифта. В ней можно передвигаться вверх и вниз, но вверх Вы не пройдете выше 3-го этажа - что-то мешает. Когда же он залез на самый верх и посмотрел вверх, то увидел дно лифта! Обследовав лифт, он нашел ниши, где и взял *GEM*.

4. Найден второй *GEM*. Он лежал в детской кровати вместе с формулой. Причем, пока Вы не возьмёте формулу, которую можно заметить только обследовав кроватку, драгоценность Вы не найдёте. Но формулу не даёт SAND-MAN (ещё один из супер-героев MARVEL). Он чувствует, что Вы ходите по песку в его комнате и тут же появляется. Выход из положения очень простой. Надо прыгнуть вверх, и оказаться на потолке.

5. Удалось найти и третий *GEM* на втором этаже, прыгнув вверх в комнате, где можно увидеть себя. Пройдя в воздухоочиститель сквозь решетку, можно по пасть к вентилятору. Передвигаясь вниз, можно увидеть MYSTERIO. Его также можно увидеть, прыгнув, когда увидите себя.

6. Обследовав аквариум водяного (HYDROMAN), Вы найдете в нем *GEM*. Но взять его не удается, так как он не появляется на экране как предмет.

7. MADAM WEB дает подсказки по команде "SCAN".

LIZARD – CaCO3+HCl=CaCl2 - смешать вещества.

MYSTERIO - IT MAPS ODD - не понятно. (*Возможно, перевод здесь: "У него странная судьба" или "Его будущее туманно" - прим. ред.*) SAND-MAN - YOU'RE ABOVE HIM (ты над ним) - прыгнуть на потолок.

HYDRO-MAN - SOLID (твердый) - не понял. (*Вполне возможно, что Мадам Веб, намекает на одно из состояний воды - твёрдое т.е. лед - прим. ред.*)

О своих проблемах в игре **MYSTERIOUS ADVENTURE** фирмы "DIGITAL FANTASIA" поведал нам Павел Заикин из г. Сургут-23. Вот как он описывает свои приключения:

Я обнаружил корабельный журнал (SHIP'S LOG), и в нем была запись "MA.Y .EL.STE .OV 18.1". Не знает ли кто-нибудь, что эта надпись значит?

(На наш взгляд, это напоминает название корабля, которое вполне могло бы выглядеть так: "MARY CELESTE NOV 18.1". Последние слова очень похожи на дату - Ноябрь 18?1 года, но проверить свою версию мы не можем, т.к. у нас нет в наличии данной программы - ред.)

В этой же программе я нашел предмет "POLICE BOX", но что это такое и как его применить, я не знаю.

Кроме того, в Древнем Египте есть потайной ход возле сфинкса. Войти туда я могу, а вот выйти - нет.

И ещё, в программе KENTILLA я набрал 28%, но в чёрной башне не могу пройти Большого Тролля, который что-то там охраняет.

Клопков Сергей спрашивает, как в игре SEAS OF BLOOD попасть в порт, сойти на землю? На месте кораблекрушения он нашел напиток, но взять его мешают спруты. Как быть?

В дополнение к прошлому "перекрестку" пришло письмо от Тверчан - Сергея Графа и Владимира Турчанинова. Они описывают свои успехи в игре EMERALD ISLE. Мы приводим здесь только те из них, которые не были описаны ранее. Итак:

- Парашют надо отдать швее;
- Топор можно найти в лесу, попасть в который можно, пройдя пещеру;
- Имея топор и руководство (MANUAL) можно построить лодку (CANOE) из KNOTTY LOG;
- К лодке можно привязать якорь (TIE ANCHOR).

В свою очередь читатели просят помощи в играх: ROBIN OF SHERLOCK-1, SEAS OF BLOOD (как начать?), APACHE GOLD (что делать в лагере?), MAGNETIC MOON-1, REBEL PLANET (как проникнуть в канализационный люк?).

Дмитрий Иванов из г. Новокузнецка делится своими открытиями в целом ряде адвентюрных игр:

RED MOON: В первой комнате дайте команду "EXAM" и Вы найдете "CROWN". Найдите рукоятку (HANDLE) и в комнате, где она находилась, вставьте её в квадратное отверстие (SQUARE HOLE). Вы откроете шлюзы, и из озера выйдет вода. Чтобы пройти METALL ROOM, в которой по ногам бьёт ток, надо надеть WELLINGTON BOOTS.

К программе надо обязательно составить карту - без неё не обойтись. Вам встретится множество противников и друзей… Не торопитесь принимать решения. Просто ходите и анализируйте (*это принцип глобальной ADVENTURE - см. статью "Визит в Level 9" в ZX-РЕВЮ № 5 - прим. ред.*)

BORED OF THE RINGS: Из первой комнаты Вы никак сами не выйдете, но после двух ходов Вас оттуда вышвырнут. Затем идите в дом дяди BIMBO. Там Вы получите задание, кольцо и охрану из трех Боггитов. Если Вы это не сделаете, то после 15 ходов Вас убьют. Сходите к фермеру и возьмите эль (ALE). Пройдите в самую восточную комнату, где дорогу загораживает блок. Подлезьте под него (DOWN). Теперь Вы можете ходить по лесу, пока не попадётесь дереву. Как только это произойдет, крикните "помогите!" (SAY HELP). Сходите к своему спасителю в гости, за это получите "MAGIC BEANS". Теперь идите на постоялый двор (INN), там Вы встретите Артифакса (ARTIFAX). Он что-то пытается сказать, но его затыкают Боггиты. Он так и будет ходить за Вами, пытаясь что-то произнести. Затем идите по мосту на восток, там встретите целую толпу друзей. Здесь Дмитрий и застрял - идти дальше некуда.

REBEL PLANET: Идти в шлюз, там сказать лакею "помоги" (SAY TO VALET HELP). Время полёта можно пролежать в гидропостели, слушая музыку (LISTEN MUSIC). Но после двух-трех периодов сна надо сходить поесть. (Вставьте визитную карточку в DISPENSOR).

Интересуется игрой REBEL PLANET и Алексей Хамардюк. Он спрашивает о той, как выбраться из тюрьмы,

где стены издевательски нашептывают, что "ты здесь сгниёшь". Его сажают туда при попытке сдвинуть люк канализации или взломать OCTOPLAST DOOR. Можно ли сделать это так, чтобы не сажали в тюрьму? Как починить скутер, хранящийся в трюме корабля? И как открыть контейнеры, находящиеся там же.

Алексей также подтверждает нашу догадку о подсказке Мадам Веб в игре **SPIDER MAN**. Действительно, HYDROMANa следует заморозить, используя термостат на четвертом этаже. Он также задает вопросы: Каким образом можно отобрать *GEM* у ELECTRO? И почему взрывается ENERGY EGG? Как общаться с таинственным MYSTERIO? Также непонятно назначение зеркала на втором этаже: передвинуть, разбить или войти в него нельзя, можно только посмотреться в него.

Но вернемся к письму Алексея.

Еде вопрос - по RETURN TO EDEN, второй части трилогии "SILICON DREAMS". Как можно пройти Левиафана, который перегородил реку? На вопросы он не реагирует, а по команде "HELP" авторы советуют обратиться к буклету, которого в наличии, естественно нет. Как быть?

Интересные сведения об игре **HUNCHBACK** сообщил нам Мучкаев Максим из Самары. Вот что он рассказал:

- Я прошел большое количество локаций, нашел много книг и кинжал для обороны от охранников. Убив одного из них, я завладел мечом - более мощным оружием. В глубине лабиринта я нашел библиотекаря. Когда отдал ему все книги, он открыл люк в полу. В люке очень темно и даже со свечой туда не пускают.

Знает ли кто-нибудь, что делать дальше?

На вопрос об игре **MESSAGE FROM ANDROMEDA**, который мы опубликовали в одном из предыдущих выпусков "РЕВЮ", пришла целая серия ответов. Наиболее полную информацию дает письмо Сергея Дрянных из Красноярска:

- Панель в "MIRRORED ROOM" двигать не надо, она является своего рода замочной скважиной. В игре есть такой предмет, как "CRYSTAL ROD", так вот его надо направить на пластинку командой "POINT ROD AT PLATE". Кстати, В этой игре часто не проходили самые очевидные команды потому, что использовался не тот предлог. Во всех командах в основном используется предлог "AT".

Его дополняет Белин А.В. из г. Нижний Новгород:

- Скажу сразу, что за этой панелью стоит пьедестал, на котором находится шар. Чтобы его взять, надо сначала повернуть (TURN SPHERE, TAKE SPHERE). Эту сферу надо применить в голубой комнате.

Корреспондент также задает вопрос, не может ли кто-нибудь помочь ему достать игру STARSHIP-3, заключительную часть первой трилогии. Вот его адрес:

603074 Н.Новгород,
ул. Куйбышева, д. 49, кв. 51
Белину А.В.

На прошлом заседании "перекрестка" был поднят вопрос об игре **VALHALLA**. Возможно, письмо Алексея Каштанова из г. Дзержинск Нижегородской обл. поможет нашим читателям разобраться с этим вопросом.

Вот список команд, которые понимает программа:

LIST - список имеющихся у персонажа предметов;

WHERE - описание локаций, и выходов из нее;

WHO - имена персонажей, находящихся в локации;

WHAT - список предметов, находящихся в локации;

HELP - иногда выдает подсказки, но в большинстве случаев пишет сообщение "SORRY…"

HOW - выдается Ваше состояние на данный момент.

В этой игре все можно продавать и покупать, были бы кроны (CROWNS), а их вначале целых 200. В связи с этим появляются новые глаголы: SELL (продать) и BUY (купить). Например:

SELL HELMET TO ODIN (продать шлем Одину).

Персонаж может согласиться (ODIN ACCEPTS), тогда он возьмет вещь и заплатит Вам деньги. Но он может также отказаться (ODIN REFUSES).

Также Вы можете покупать что-либо у персонажей:

BUY SHIELD FROM THOR WITH 20 CROWNS. (Купить щит у Тора за 20 крон). Персонаж может согласиться или отказаться.

Но неясным остается главный вопрос, что же нужно сделать в этой игре? Есть также вопросы по другим адвентюрам:

LORD OF THE RINGS-2. Что надо сделать в LOTHLORIEN'e, чтобы Гимли не убивал посланника эльфов после того, как они переходят реку. А то доходит порой до абсурда: приказываете Гимли оставить топор в соседней локации, а когда эльф пытается завязать ему глаза, то программа выдает сообщение, что Гимли убил эльфа топором, хотя никакого топора у него нет. Если же после этого попробовать углубиться в лес, то эльфы быстренько убивают всю компанию.

А вот в игре **VERA CRUZ** Алексею удалось разузнать, как работать с опцией "MESSAGE". Оказывается, с её помощью можно посылать сообщения в различные учреждения:

PRIS - тюрьма;

PREF - префектура;

GIE - жандармерия (дело происходит во Франции);

GIAT - ??? Расшифровать

GRRJ - ??? не
BDRJ - ??? удалось

Полностью пройти игру **EXCALIBUR** удалось Александру Дешкову из Челябинска. Он прислал нам список основных подсказок к этой игре:

LEAN LADDER	BREAK TRAPDOOR
BREAK LADDER	LIGHT LAMP
TIGHT GAP	THROW SALT
EXAM BUSH	OIL WINCH
EXAM LOGS	TURN WINCH
KILL KNIGHT	MIRROR SPELL
UNLOCK DOOR	WAKE KING
BUY LAMP	

Остальное - дело техники, и Вы станете Рыцарем Круглого Стола.

Также он прошел игру **TIME QUEST**. В ней надо набрать семь сокровищ и найти три магических слова. Подсказки к первому этапу игры очень незамысловаты:

FEEL GRASS	FEEL CRACK
PUSH BOULDER	LIGHT LAMP
SLIDE TABLE	DIG

После того, как Вы найдете два сокровища, нужно бросить (THROW) Веревку на утёс (CLIFF) и дать команду CLIMB ROPE. Потушите лампу (OUT LAMP) и уже после этого потрите её (RUB). Появится джинн и сообщит Вам волшебное слово. Просто наберите его, и он перенесёт Вас в новую область. Завершается программа, похоже, но тоже с некоторыми хитростями (нужно встать на волшебный ковер).

По программе "WORM IN PARADISE" фирмы "LEVEL 9" задает серию вопросов команда Raven Group из г.Владивостока:

- как зарегистрироваться для получения работы?
- как купить билет, чтобы переправиться через реку?
- как связаться с социалистами - в явочной квартире никого нет.

Интересное наблюдение прислал нам А.Н. Орлов, живущий в Воронежской области:

В книге Роджера Желязны "Хроники Эмбера", нашел знакомое название - TIR-NA-NOG'TH. Вот фрагмент из текста:

"Мы считаем,- продолжала Дейдра,- что только Лабиринт вернёт принцу Корвину память. Он не может пройти его в Эмбере, а следовательно, остается Рембэ, потому что в Tir-na-nog'th, как ты понимаешь нам не попасть…"

Ко всему прочему, если героем игры является Cuchulian, то в книге есть персонаж Джулиан, один из пятнадцати братьев. Я подумал, вдруг игра TIR-NA-NOG основывается на этой книге. Может быть, кто-то знает точно?

Комментарий ИФК: Нет, игра никак не основывается на этой книге, хотя общие корни, конечно есть. Дело в том, что Роджер Желязны (как и большинство авторов, работающих в жанре "фэнтэзи") часто использует в своём творчестве мотивы из древних легенд или мифов. В данном случае просто авторы игры и Р. Желязны вдохновлялись одним ирландским и кельтским фольклором.

В ZX-РЕВЮ № 4 в разделе "Форум" прозвучало несколько призывов к помощи. И помощь приходит. Михолап А.Н. (DEVILMAN) из Тулы отвечает на вопрос "неизвестного поклонника Speccy из С.Петербурга", застрявшего в **LORD OF THE RINGS-1**.

Если не задерживаться долго в норе, то после посещения дома MAGGOT'a на пристани Вы встретитесь с MERRY, который приплывет на пароме (FERRY). Если вы все-таки опоздали к его (MERRY) прибытию, то есть способ переправиться через реку на пароме. Для этого нужно покрутить ручку на пристани (TURN HANDLE), пока не появится паром, однако, ручка тяжёлая и с трудом крутится, поэтому, чтобы не выбиться из сил, которые не бесконечны в этой адвентюре, нужно крутить ручку по очереди. Не исключено, что придется все же пополнить силы, съев чего-нибудь. Кстати, чтобы заставить двигаться паром (не забудьте влезть на него: BOARD FERRY), нужно опять крутить ручку, но уже ту, что находится на пароме. Ещё одно замечание: ручку (HANDLE) можно взять с собой, но где она понадобится, и понадобится ли вообще, придется узнать Вам самим!

В принципе, необязательно перебираться через реку на пароме - это потребует много сил, но этот путь - самый безопасный. Есть ещё путь через мост, но он охраняется Чёрными всадниками; менее опасный путь - длиннее (тот, что протянулся вдоль реки).

Ещё несколько советов по первой части игры. Как только программа загрузилась, задается вопрос: "Какими персонажами Вы хотите управлять?". Здесь нужно ввести через запятую имена тех хоббитов, которыми будете управлять Вы, остальными - компьютер. В процессе игры можно переключаться между теми хоббитами, которые были выбраны вначале. Обязательно выбирайте Фродо и Мерри, чтобы вся команда была в сборе. Здесь есть свои плюсы: чтобы управлять персонажами (хоббитами), теперь можно переключиться на того, кто Вам нужен, лишь введя его имя, вместо того, чтобы набирать длинное "SAY TO … <действие>". Иногда бывают случаи, когда персонаж, получив команду, игнорирует её, и приходится давать её снова.

Другой плюс: Появляется возможность производить разведку (благоразумнее всеми, кроме самого Фродо, ведь он - хранитель кольца.)

Ещё один плюс: Если у компьютера собралось несколько любителей адвентюр, то они могут играть в "LORD OF THE RINGS" вместе, взаимодействуя или противоборствуя между собой. Последнее, хоть и является отклонением от сюжета, но все равно интересно.

Минус: игра становится чуть длиннее.

И ещё. Хоть в **LORD OF THE RINGS-1** я ушел далеко, все-таки закончить первую часть мне не удалось. Причина - GREEN KNIGHT. Может быть, кто-то из читателей знает, что нужно делать, чтобы получить BROKEN MEDALION? Есть подозрение на CANNABIS SATIVA (цветок).

Совет москвичу Юрию Васину по адвентюре **RETURN TO EDEN**. Обитатели космического корабля "SNOWBALL", одним из которых являетесь и Вы, несправедливо решают, что Вы - предатель Вы покидаете корабль и высаживаетесь на планету EDEN. С этого момента и начинается игра. Чтобы избавится от смерти вначале игры, найдите нору крота (MOLE-HILL), а там - ориентируйтесь сами. Кстати, смерть происходит из-за того, что корабль SNOWBALL поворачивает свои дюзы к планете и стартует в космос. Найдите способ противодействия или бегите подальше.

Максим Полулях из Киева интересуется адвентюрными играми и задает несколько вопросов: Где в игре **HEAVY ON THE MAGIC** найти рубин? Как отдать три предмета колдуну в программе SPIKE IN TRANSILVANIA?

Что-ж, Максим, мы всегда рады помочь. Рубин (RUBY) находится на четвёртом этаже замка внизу слева. Он выглядит как обычный кусок скалы, поэтому его можно не заметить. Пройти в комнату с рубином и дверью можно с помощью демона Астарота или с помощью ERLSTONE сквозь отсутствующие ворота.

В программе **SPIKE IN TRANSILVANIA** колдун ничего не просит ему принести, а советует смешать в его котле сок жабы, раненую летучую мышь и книгу заклятий. Все это надо бросить в котёл и Вы получите галстук, с которым Вас пустят в ресторан при гостинице.

Кроме того, Максим просит опубликовать его адрес для переписки со всеми любителями "SPECTRUM". С удовольствием выполняем его просьбу.

253225, Украина, Киев,
ул. Каштановая, д. 15-Б, кв. 17
Полулях Максим

И снова нам пишет В.А. Исаев из г. Владивостока. Он прислал нам целый каскад вопросов по адвентюрным играм. Ну, что ж, всё по порядку:

1. Игра RIGEL'S REVENGER. Если нас процитировать, то "самое сложное - это начать игру". Как же это сделать?

2. HAMPSTEAD - как купить билет на поезд? Можно ли обменять UB40 на наличные или как им заплатить в магазине?

3. ROBIN OF SHERLOCK. Когда я вхожу в поезд - он не едет и отправляется только тогда, когда я выхожу. Как мне уехать на поезде?

4. SLANEKING - с помощью, каких предметов можно воспользоваться кузницей? Как набрать воды в ведро?

5. KAYLETH - как же в самом деле начать игру? Ни BREAK ни BEND не действовали ни до Вашего совета, ни после.

- Может быть, попробовать ввести "BREAK BEND"? Вдруг что-нибудь получится…

6. STARSHIP-1. Проблема в том, что у меня только две минуты на переноску вещей в шлюпку и отлет. За это время можно сделать только одну пробежку "склад-шлюпка", а взять разом все предметы невозможно. Как быть?

7. LORDS OF MIDNIGHT. Сражения с армиями Тьмы ни к чему не приводят, а найти ICE CROWN я не могу. Где её искать?

И ещё два шедевра LEVEL 9. Оба они, несомненно, заслуживают "мозгового штурма", и я хочу предложить их в качестве следующих "осажденных".

Это RED MOON - игра, с которой можно разобраться, так как в одиночку я прошел уже 33% игры, но всё же есть вопросы: Как вскрыть сейф в сокровищнице? Как поднять решетку, преграждающую доступ к монетам? Как избавиться от призраков убитых врагов, которые возникают на каждом шагу, что сводит игру к цепи бесконечных сражений. Как, наконец, реально увеличить свою магическую силу, чтобы использовать заклинания?

Второй шедевр - **WORM IN PARADISE**. Сюжет игры описан в сборнике "500 игр № 3" издательства "VA PRINT".

С этой игрой у меня - полный пролёт. Я, конечно, прошелся по городу, собрал коллекцию различных предметов - пицца, газета, пирог, флаг и т.д. Но что мне далось выяснить реально?

1. Перед арестом, потенциальный социалист оставил бумажник с цветовым кодом у памятника Третьей Мировой войне. Цветовой код обозначает месторасположение квартиры. Побродив по транспортной системе, я добрался до квартиры социалиста, но ничего и никого там не нашел.

2. Сдав бумажник и код социалиста полиции, я получил новое место жительства - видимо в награду.

3. Две команды транспортной системы:

SAY "HOME" - двое дроидов отнесут Вас домой.

SAY "EXIT" - к выходу транспортной системы.

4. Цена DOGGETA, робото-собаки, вначале составляет 900CR, но если подождать пару-тройку дней, её снизят до 600-700CR. Однако если ждать слишком долго, то её продать откажутся.

5. В кубке, который находится в музее, Вы найдете CUSTARD нечто вроде загустевшего сладкого крема (?).

В игре вообще много подозрительных моментов: какие-то трещинки в стенах, которые напоминают потайные

ходы; непонятная комната в кинотеатре и т.д. и т.п…

Илья Разумовский из г. Пермь задает вопрос по игре **HAMPSTBAD**.

- После опубликования в ZX-РЕВЮ информации об этой программе, я увлекся этой интересной игрой. Но, не имея достаточного опыта работы с жанром ADVENTURE, я "застрял" в самом начале.

Я вышел (вернее, выехал) из своего дома и теперь должен, согласно данному Вами предписанию, вступить в клуб и устроиться на работу, но ни то, ни другое мне никак не удается. На мой вопрос отвечают, что работы нет. Также не знаю, куда пристроить UB40 (ничего не могу сделать в SOCIAL SECURITY OFFICE, где, по-видимому, должны выдать пособие). Помогите, если можно.

Вот и всё, мы прощаемся с Вами до следующего заседания "перекрестка".

*** * ***

В прошлой выпуске ZX-РЕВЮ мы начали рассказ о системе правил AD&D, на базе которой строятся многие ролевые игры. И пообещали, что рассмотрим этот вопрос кратенько в двух номерах журнала.

Даже первый наплыв откликов превзошел, наши самые смелые ожидания Идея показалась читателям интересной и многообещающей. Нас просят не останавливаться и продолжить работу в этом направлении. Идя навстречу многочисленным пожеланиям, мы пересмотрели свои первоначальные намерения и готовы серьёзно расширить первоначальные планы и уделить этому вопросу не пару статей, а столько, сколько понадобится для того, чтобы хотя бы кратко, но по-возможности всесторонне его осветить. Так что, если наши статьи Вас заинтересовали, вы сможете продолжить их чтение их и в ZX-РЕВЮ-95.

*И ещё… Опять же, учитывая повышенный интерес, мы решили отделить мух от котлет и отдельно повести рассказ о системе **AD&D** и разговор о структурах данных в ролевых играх (**ПРОЕКТ RPG**). Так что вместо одного нового цикла Вы получите целых два.*

Система правки AD&D

© Сергей Симонович, 1994.

Продолжение. Начало см. в ZX-РЕВЮ № 5, стр. 70.

5. Основные режим игры.

Герои игры в любой момент времени могут находиться только в одном из двух возможных режимов - Adventuring Mode и Combat Mode. Если Вы не новичок в компьютерных играх, то понимаете, что режим Adventuring соответствует путешествию по игровому пространству, перемещению от локации к локации, изучению географии и топографии игры.

Режим Combat - боевой режим. В него игра переходит, когда в одной локации встречаются PC (Player's Characters - управляемые герои) и NPC (Non Player's characters - неуправляемые персонажи, а попросту говоря, монстры).

Эти два режима, как правило, не пересекаются, то есть те действия, которые разрешены Вам в режиме Adventure (поспать, поесть, посмотреть, побродить, потренироваться, починить оружие, подлатать доспехи и т.п.) будут запрещены в боевом режиме Combat Mode. Там Вам остается только наносить удары, парировать их, менять оружие, глотать исцеляющие таблетки и эликсиры, а от всех способов перемещения в игровом пространстве остается только один - убежать, да и то, если противник Вам позволит. Единственный случай, когда эти режимы могут сосуществовать одновременно - это когда производится дистанционная атака метательным оружием.

6. Adventure Mode.

Как и всё в ролевых играх, перемещение героев в игровом пространстве может быть магическим, а может быть и обычным, естественным. Мы сегодня не будем останавливаться на магических вещах и вернёмся к ним в одном из последующих выпусков журнала. Магия - это особый большой и сложный вопрос и им надо заниматься отдельно. А из естественных способов перемещения в игре нам остаются: Ходьба, Бег, Перемещение верхом на животном. Плавание, Плавание на судне, Лазание по веревкам, лестницам, скалам и, пожалуй, всё. Не всё мы можем здесь рассмотреть, но некоторые основные положения отметим.

6.1. Скорость перемещения.

Сразу скажем, что это понятие - ОПЦИОННОЕ, т.е. оно может быть задействовано в игре, а может и нет. А если Вы сами пишете игру, то делайте, как Вам нравится.

6.1.1. Ходьба.

Концепция путешествия в пространстве тесно связана с ресурсом времени. Если в Вашей игре идет учёт времени, то необходим учёт скорости перемещения героев. Для основных рас персонажей принято следующее соотношение:

Люди	– 12
Полуэльфы	– 12
Эльфы	– 12
Пещерные Гномы	– 6
Лесные Гномы	– 6
Хафлинги	– б

Эти данные соответствуют скорости ненагруженных героев. Если же им приходится нести груз, (а во многих играх это учитывают), то скорость героя понижается до тех пор, пока он вообще не сможет сдвинуться с места. В компьютерных играх можно учитывать и усталость, и ранения, и болезни, и возраст для оценки снижения скорости.

Скорость же всего Вашего отряда берётся по средней скорости героев (сильные помогают слабым). Сама величина скорости пользователю постоянно не докладывается, но возможность посмотреть её у него имеется. Зато ему постоянно сообщают о смене дня и ночи или о смене суток. Если герои перегружены и изранены, то после каждой пары шагов программа говорит, что "вот и ещё день прошёл". А здоровые и легкие герои могли бы за сутки пройти локаций 10-20.

Есть особенность при путешествии по лабиринтам, замкам и пещерам. Ведь если на открытом воздухе Вы можете за пару десятков шагов пройти путь между двумя городами, озерами, горами, то в лабиринте за столько же шагов Вы пройдете только малую его часть. Обычно просто предполагается, что там Вы идете в 10 раз медленнее, внимательно рассматривая все на своем пути и прислушиваясь к каждому шороху.

6.1.2. Форсированный марш.

Обычным шагом не перегруженный человек может идти до 10 часов в сутки и проходить до 30 км. При этом предполагается, что никакого истощения он не испытывает. Но в тех случаях, когда он хочет совершить форсированный марш-бросок ценой собственного здоровья, ему это позволено. В этом случае его скорость увеличивается в 2,5 раза, но в конце каждого дня такого пути выполняется контрольный бросок костей для проверки состояния здоровья (параметр CONSTITUTION). Эту проверку делают по среднему параметру для всех членов группы (тоже предполагается, что сильные и здоровые герои поддерживают слабых и больных). Если проверка прошла (Ваш средний параметр больше, чем результат броска костей), то на следующий день опять можно продолжить форсированный марш. Но за каждый день форсированного марша вводится штраф- -1 к текущему параметру CONSTITUTION героев, так что долго двигаться форсированным маршем Вам не удастся.

Штраф снимается только отдыхом. За каждый день форсированного марша надо заплатить полуднем отдыха, пока здоровье не восстановится до исходного уровня. Если Вы не восстановите здоровье, а наткнётесь на противников и вступите в бой, то будете нести штраф в размере по -1 к величине силы своего удара за каждый недостающий балл здоровья. Но об этом мы вспомним позже, когда будем рассматривать Combat Mode. Здоровье может восстанавливаться и целительной магией или лекарствами (об этом ниже), но отдыхом можно восстанавливать только здоровье, подорванное маршем. Болезни или магический подрыв здоровья обычным отдыхом не восстановить - нужны другие меры.

Форсированный марш возможен только на открытом пространстве. Ни в замках, ни в лабиринтах, пещерах, горах, болотах, лесах и т.п. так двигаться нельзя.

6.2. Плавание.

Все персонажи либо умеют плавать от рождения, либо могут обучиться этому искусству. Кстати, если герой может управлять судном (имеет навык морехода) это ещё не означает, что он умеет плавать. Если все Ваши герои обучены плаванию, то Вы можете перемещаться по спокойной воде со скоростью вдвое меньшей обычной скорости ходьбы. К спокойной воде относятся реки и малые водоемы. В больших озерах и морях к спокойной воде относятся только прибрежные поля (смежные с сушей).

Если игра проходит в большом окружении водных пространств и мотив плавания и мореплавания является важнейшим, то в игре вводятся разные степени обученности, производится учёт усталости в зависимости от расстояния, учитывается скорость плавания, уровень героев и многое другое.

6.3. Вертикальное лазание.

Только ВЗЛОМЩИКИ имеют природные способности к лазанию по веревкам, скалам, стенам и т.п. Остальным персонажам надо этому учиться, и существуют три степени подготовленности персонажей к этому делу:

- профессиональный ВЗЛОМЩИК;
- подготовленный альпинист;
- нетренированный персонаж.

Независимо от степени подготовки только ВЗЛОМЩИК может взбираться без веревки и без лестницы по каменным стенам и другим, гладким и скользким поверхностям. Альпинисты могут это делать только с соответствующим снаряжением. Нетренированные персонажи тоже могут подниматься, но лишь в присутствии альпинистов (их поднимут на веревке).

Этот способ передвижения происходит путем оценки шанса успешного подъёма с помощью расчётных таблиц. Ниже мы приведем пример двух таких таблиц:

Таблица 4.

Вероятность успеха вертикального подъема.

ВЗЛОМЩИК, обученный альпинизму	(уровень*10% + 10%)
ВЗЛОМЩИК	уровень*10%
Обученный альпинист	40% + 10%*едницы навыка.
Необученный персонаж	40%

Это базовая таблица, на которую накладываются дополнительные модификаторы, зависящие от прочих факторов (см. табл.5).

Таблица 5.

Ситуация	Модификатор
Наличие опоры (ветви деревьев, деревянные и верёвочные лестницы)	+40%
Стена + верёвка (верёвка закреплена и опущена вниз)	+55%
Поверхность наклонная	+25%
Доспехи	
панцирь и т.п.	-25%
металл, доспех	-50%
кольчуга и т.п.	-15%
кожаные и набивные доспехи	-5%
Раса персонажа	
пещерные гномы	-10%
лесные гномы	-15%

хафлинги	-15%
Груз (*)	-5%*(x)
Состояние поверхности	
Влажная	-25%
Скользкая	-40%
Здоровье	
персонаж рамен	-10%

(*) Примечание: интересна система учёта груза. Это делается через учёт скорости обычного движения. Поясним это на примере. Пусть скорость человека без груза равна 12 пунктам. Пусть его скорость с грузом равна 8 пунктам. Снижение за груз - 4 пункта. Эту величину умножают на 5% для расчёта вероятности успешного подъёма на вертикальное препятствие.

После того, как все данные из табл. 4 и 5 рассчитаны, производится бросок 100-гранной кости. Если результат выше, чем полученное критическое значение, попытка не получилась, а если ниже, то всё O.K.

Рассмотрим пример. *Ваш гном необученный для скалолазания отстал от отряда и оказался перед высокой скалой. Вероятность того, что он сможет её преодолеть составляет 25% (40% - 15%). При броске костей выпало число 44. Попытка преодолеть препятствие окончилась неуспешно. Вся команда потеряла ход. К счастью, кому-то из персонажей, уже взобравшихся на скалу, пришла в голову мысль спустить веревку (+55%). Теперь вероятность успеха .- 80%. Кости показали число 73 и попытка закончилась благополучно.*

В настольных играх вертикальному лазание уделяют большое внимание. Даже находясь на восхождении, герои продолжают общаться друг с другом, применять магию и магические предметы. Здесь учитывается и скорость восхождения, и усталость и сила и здоровье. В компьютерных играх использовать все параметры и предусмотреть всё многообразие ситуаций, которые могут возникнуть при вертикальном восхождении, просто невозможно. Там обычно все проще. Обучены герои скалолазанию - значит, могут ходить по клеткам, на которых расположены горы. А если не обучены - то либо нечего им тут делать, либо пусть ищут деньги для оплаты за обучение.

То же и с форсированием вертикальных стен. Если обычный вход в замок закрыт, то для того, чтобы забраться на стены, надо иметь веревки, лестницы и т.п. А если ничего этого нет, то… "до свидания, приходите в другой раз, когда будете лучше подготовлены".

7. ОПЫТ.

Опытность героев - один из важнейших параметров в ролевых играх. Поскольку основная цель игры состоит в воспитании героев, то можно сказать, что набор опыта - одна из основных задач, стоящих перед играющими. Опыт непосредственно связан с уровнем подготовленности, а от последнего зависит очень многое: доступность тех или иных магических заклинаний, сила удара, способность защищаться и многое-многое другое.

7.1. Групповой опыт.

Пункты опыта (Experience Points) зарабатываются героями за их действия по освоению и изучению игрового пространства. Так, за преодоление физических или логических препятствий, за решенные по ходу игры головоломки вся группа получает пункты опытности. В этом случае все члены группы получают равные баллы, независимо от того, какое участие в событии принял тот или иной герой (предполагается, что все успехи обеспечиваются только коллективными действиями). ВОЛШЕБНИК, простоявший всю битву в стороне с готовым к применению заклинанием, которое оказалось без необходимости, имеет не меньше прав на призовые баллы, чем ВОИН, который нанёс решающий удар сопернику.

7.2. Индивидуальный опит.

Но герои могут получать и индивидуальные призовые пункты за те или иные события. В этом случае они получают их в соответствии со своим амплуа. Опять вернёмся к бою. Если он прошел врукопашную, то кроме группового поощрения для всех членов команды будут даны дополнительные баллы ВОИНУ, и чем тяжелее была битва, тем больше баллов он получит.

ВОЛШЕБНИК повышает свой опыт после удачных применений своих заклинаний. Если во время блуждания по лесу он будет метать магические громы и молнии направо и налево, то не получит ничего, но если в тёмном подземелье он осветит путь группе, то получит баллы.

СВЯЩЕННИК повышает свой опыт тоже в результате удачного применения магии, но его магия, как мы знаем, целительная и защитная. Как и ВОЛШЕБНИК, он повышает индивидуальный опыт за изученные заклинания и прочитанные свитки.

ВЗЛОМЩИК набирает индивидуальные баллы за взломанные двери, открытые сундуки и т.п.

В компьютерной игре реализуют иногда такой приём. Перед группой стоит некая задача, и программа спрашивает, кому Вы хотите её поручить (например, войти в бассейн с серной кислотой и посмотреть, нет ли чего в нем интересного?). Вы выбираете добровольца на это дело. Если поиск кончится успешно,- он получит баллы опыта, а если нет, то… расплатится собственным здоровьем, а то и жизнью.

В настольных играх Dungeon Master может время от времени премировать индивидуальными баллами опыта тех персонажей, которые, на его взгляд, хорошо вошли в роль и красиво интерпретируют своих персонажей. О необходимости вживания в роль мы говорили в прошлом выпуске ZX-РЕВЮ.

7.3. Тренировка.

Набор героями опыта является необходимым, но не достаточным условием для повышения уровня. Как правило, для того, чтобы превратить баллы опыта в более высокий уровень, надо пройти тренировку. Нередко за

тренировку приходится платить деньги и, конечно, на неё расходуется время (обычно несколько недель).

Поскольку игры бывают с очень большими географическими пространствами, когда для прохода от одного региона к другому необходимы сотни ходов, то в них встраивают несколько тренировочных центров. Накопив опыта и денег группа идёт к ближайшему тренировочному залу. Обычно в играх делают так, чтобы разные тренировочные центры занимались подготовкой для разных уровней. Так, ближайший к началу игры центр занимается тренировками для уровней 1…10, следующий - 11…20 и т.д.

8. Combat Mode.

Итак, Вы наконец-то встретились с группой противников, и начинается схватка. Как правило, подобные схватки в реальной жизни происходят весьма беспорядочно, но здесь в игре мы должны подчиняться единым правилам и нам следует подумать о порядке.

8.1. Раунд.

Схватка состоит из раундов. Считайте, что раунд - это как бы один ход. Только одно действие можно сделать в течение раунда, как бы просто оно ни было - нанести удар, парировать его, отступить, сменить оружие, произнести заклинание, глотнуть исцеляющего эликсира… - только одно действие.

В компьютерных ролевых играх раунды могут быть как в тактовом режиме, так и в режиме реального времени. Проще играть, конечно, в тактовом режиме. Вы можете не спеша подумать, кто кого и чем будет бить. В режиме реального времени, пока Вы думаете, компьютер будет наносить Вам удар, за ударом.

8.2. Порядок ходов.

Мы будем рассматривать только игры, идущие по тактовому принципу. Настольные AD&D всегда идут в таком режиме (если Вы своей медлительностью не рассердите Мастера и он не накажет Вас пропуском хода). Большинство компьютерных игр также работают в тактовом режиме. Режим реального времени скорее подходит аркадным играм, а не RPG.

И здесь важным моментом становится выбор последовательности ходов - кто первым наносит удар.

В настольном варианте DM бросает за каждого участника схватки кубик 1d10 и тот, у кого выпадет меньше, будет действовать первым. Далее в порядке возрастания.

Порядок обмена ударами не является чем-то постоянным. Одна схватка может состоять из десятков раундов и в каждом раунде порядок может меняться. Опять же в настольном варианте игры Мастер учитывает множество дополнительных факторов и вносит коррективы в порядок ходов. Так, он учитывает усталость героя (+2), скользкую поверхность (+2), расположение на возвышении (-1), бой во враждебной среде, например под водой (+6), бой в неудобном положении, например по пояс в воде (+4), особые скоростные качества героя (-2).

В компьютерной игре решение должен принять компьютер, руководствуясь какими-то объективными критериями. Как правило, учитываемыми факторами являются скорость и ловкость персонажей. Конечно, учитывается и случайный фактор. Может быть, Вы посчитаете нужным учитывать и усталость героев.

В настольной игре часто при определении порядка обмена ударами ограничиваются расчётом для группы в целом, т.к. для каждого героя бросать кость занимает много времени. В компьютерной игре ничто не мешает индивидуальному подходу.

Ваши герои, хоть и не всегда люди, но, по крайней мере, близки к ним и они наносят удары по одному за ход. Среди противников могут быть любые монстры, в том числе и такие, которые могут наносить не по одному удару. Например, Вы не удивитесь, если трехголовый дракон нанесет пару ударов зубами и один раз дохнет пламенем за один ход. Многие магические существа наносят и большее количество ударов за ход.

В некоторых вариантах игр Вашим героям разрешается наносить по два удара за один ход, если они вооружены двумя видами оружия. Например, мечом и кинжалом. Разумеется, при этом они лишены возможности использовать щит.

8.3. Нанесение удара.

Удары могут быть трёх видов. Это удары обычным ручным оружием (или просто кулаком), удары метательным оружием и магические удары. О магии мы поговорим в другой раз, там принципы те же, только эффект другой, а сейчас рассмотрим, как рассчитывается результат обыкновенного удара холодным оружием. И начнем с того, что каждый персонаж в игре обладает таким важным параметром, как Armor Class - степень защиты.

Armor Class (AC) зависит от того, есть или нет у персонажа доспехи, броня, кольчуга и т.п. Очевидно, он "привязан" к персонажу и зависит от его оснащенности, но это не все… В некоторых ситуациях (если программист или DM это предусмотрели), на AC могут влиять обстоятельства, то есть вводится поправочный модификатор. Так, если Вы спрятались за каменным валуном, то при атаке спереди Ваш AC улучшается, зато при атаке сзади - ухудшается.

Эффект от Ваших доспехов можно понимать по-разному. Например, можно считать, что доспехи снижают степень повреждения от удара. Не у них есть ещё одна функция они могут вообще предотвратить удар. В классической системе AC следует понимать как степень снижения вероятности того, что удар достиг цели. Поэтому чем лучше защита, тем меньше параметр AC. Так, в случае отсутствия доспехов, AC=+10, а при самой наилучшей (самой мощной магической защите) AC=-10. Таким образом, чем меньше параметр AC - тем эффективнее защита.

Кроме доспехов на этот параметр влияют щиты, шлемы и даже такая мелочь, как сапоги, перчатки и т.п.

Личные способности также могут влиять на класс AC. Так, например, при высокой ловкости героя, вводится поправочный модификатор, улучшающий AC.

Параметр AC настолько понятен и естественен, что перекочевал из ролевых игр и в другие жанры. Так, например, в любимой народом Laser Squad когда Вы выбираете себе броню, Вы тоже платите деньги за Armor Class.

Справочные таблицы содержат информацию о десятках видов оружия. Если захотите сделать собственную игру, составите себе такие таблицы, какие пожелаете.

Это то, что касается защиты. С другой стороны, каждый персонаж способен по-разному наносить удары. Чем выше его уровень, чем больше его сила, тем мощнее его удар. Способности к нанесению удара выражаются параметром с труднопроизносимым названием THAC0.

THAC0 - это буквосочетание можно расшифровать как "To Hit Armor Class 0". Грубо говоря, этот параметр, присущий каждому персонажу, показывает, сколько должно выпасть на костях (To Hit), чтобы его удар достиг цели при условии, что Armor Class противника равен нулю. Зависит этот параметр, как от амплуа нашего персонажа, так и от его уровня. Чем больше сила героя, тем выше THAC0. С потерей здоровья, с ранениями THAC0 может падать.

Примечание: в компьютерных играх параметр THAC0 очень часто проставляется как просто TH.

В таблице 6 мы привели данные о том, как влияет амплуа и уровень героев на их THAC0.

Таблица 6.

уровень	Воин	Священник	Взломщик	Маг
1	20	20	20	20
2	19	20	20	20
3	18	20	19	20
4	17	18	19	19
5	16	18	18	19
6	15	18	18	19
7	14	16	17	18
8	13	16	17	18
9	12	16	16	18
10	11	14	16	17
11	10	14	15	17
12	9	14	15	17
13	8	12	14	16
14	7	12	14	16
15	6	12	13	16
16	5	10	13	15
17	4	10	12	15
18	3	10	12	15
19	2	8	11	14
20	1	8	11	14

8.4. Атакующий бросок.

В основе всей системы боя лежит "атакующий бросок". Это бросок кости, показывающий прошёл удар или нет. Число, которое должен выбросить игрок, чтобы удар считался состоявшимся, называется числом TH (To Hit). Атакующий бросок применяют во всех случаях, когда есть сомнение, что удар по пал в цель, причем это не зависит от вида удара. Это может быть удар кулаком, мечом, бросок камнем, выстрел из лука и т.п.

Параметр To Hit рассчитывается по простой формуле: из параметра THAC0 атакующего вычитается класс защиты обороняющегося:

```
TH = THAC0 – AC
```

Если AC имеет отрицательное значение, то он, естественно, не вычитается, а прибавляется.

Теперь выполняется бросок двадцатигранной кости. Если полученное значение равно или больше, чем рассчитанное значение TH, то удар достиг цели.

Пример: *Ваш герой имеет амплуа ВОИН и достиг 7-го уровня. По таблице находим его параметр THAC0. Он равен 14. Это значит, что вам надо выбросить 14 или больше, чтобы ударить противника с защитой AC=0. В бою вам встретился орк, защищенный кольчугой (AC=6), поэтому Вам достаточно выбросить 8 (14-6=8) очков, чтобы его поразить. Если Вам это удалось, то производится ещё один бросок кости, по которому определяется степень повреждения.*

Всё вроде бы просто, но это только пока. В реальной игре дополнительно действуют разные модификаторы, учитывающие Вашу силу. Ваше оружие и т.п.

8.5. Модификаторы TH.

Влияние силы (Strength). Чем сильнее герой, тем больше вероятность, что его даже и не очень точный удар нанесет противнику урон. В таблице 7 приведена выдержка из справочников. Этот модификатор прибавляется к THAC0.

Таблица 7.

Strengh	Модификатор TH
1	-5
2	-4
3	-3
4-5	-2

6-7	-1
8-16	0
17-18	+1
19-20	+3
21-22	+4
23	+5
24	+6
25	+7

Влияние магии. Поскольку в игре большое участие принимает магия, её тоже надо учитывать. "Зачарованное" оружие повышает вероятность нанесения удара на +1. И наоборот "проклятое" оружие уменьшает эту вероятность на -1. Заклятья и проклятья на оружие накладывают маги и можно сделать так, чтобы эти модификаторы были не строго равны плюс или минус единице, а зависели бы от магических способностей лица, наложившего заклятье.

От "проклятого" оружия лучше избавиться любым доступным способом. Если нет возможности снять злые чары, то лучше его выбросить.

Соответствие оружия и зашиты. Конечно, в мире нет абсолютного оружия и абсолютной защиты. Для каждого вида атакующего оружия есть своя наиболее подходящая защита. Оружие условно разделяют на три типа: режуще-рубящее, колющее, ударное. Например, меч - рубящее оружие, копье - колющее, а булава - ударное. Подбитый войлоком кожаный доспех лучше защищает от удара палицей, чем тонкая кольчуга, зато от удара копьем кольчуга защищает лучше. Учёт этого модификатора достаточно сложен и хоть его и делают в настольных играх, в компьютерных аналогах он нам не встречался.

8.6. Граничные условия.

С учётом всех модификаторов может получиться так, что параметр TH окажется больше 20 или меньше единицы. Стоит ли бросать двадцатигранную кость, если и так известно, что TH=23 и удар гарантированно не достигнет цели?

Кость надо бросать в любом случае. Существует правило, что если на кости выпадет число 20, то атака удалась при любом раскладе, а если выпала 1, то атака безуспешна.

Так что, есть 5%-ная вероятность, что Ваш герой, как бы слаб он ни был, сумеет ударить могучего великана. Правда, надо бы ещё посмотреть, а велик ли вред от этого удара. Вряд ли удастся великана убить так просто.

8.7. Величина повреждения.

Как бы сильны и ловки Вы ни были, а рано или поздно в своих приключениях Вам придется испытать на себе удары врагов. Система учёта повреждений и ранений в ролевых играх абстрактна. До того, чтобы разбираться, что Вам отхватили топором - руку, ногу или голову, система не опускается. Весь учет идет арифметически по баллам.

Каждый персонаж, будь то PC или NPC, имеет определенное количество баллов HP (Hit Points). И когда некий тролль получает удар по голове о потерей 8-ми пунктов HP, они не относятся к его голове и не распределяются между прочими частями тела. Утрата пунктов HP происходит кумулятивно до тех пор, пока персонаж совсем не помрет или не найдет способа подлечить свои раны.

8.7.1. Раны.

Всякий раз, как в результате "атакующего броска" фиксируется попадание удара в цель, происходит оценка величины повреждения. Это опять-таки делается с помощью справочных таблиц.

Во-первых, существуют таблицы для всевозможных видов оружия. В них в зависимости от того, чем и по кому наносится удар. Вы можете найти формулу для определения величины повреждения. Конечно, немыслимо здесь печатать все, что есть в справочниках, и мы дадим только краткую выдержку.

S(mall) и M(iddle) - это когда удар наносится по жертве малого или среднего размера. L(arge) - удар по крупному противнику. Как Вы понимаете, есть разница в том, на Кого Вы идете с кинжалом - на волка или на дракона.

Таблица 8

Оружие	S	M	L
Легкая стрела	1d6	1d6	1d6
Тяжелая стрела	1d8	1d8	1ds
Боевой топор	1d8	1d8	1d8
Палица	1d6	1d6	1d3
Кинжал	1d4	1d4	1d3
Нож	1d3	1d3	1d2
Копье	1d6	1d6	1d8
Меч	1d8	1d8	1d12
Меч двуручный	2d4	2d4	2d8
Трезубец	1d6	1d6+1	3d4

Из этой таблицы интересно отметить, что есть оружие, рассчитанное на мелких и средних монстров, а есть специально для "крупного зверя". Если сами задумаете делать ролевую игру, мы думаем у Вас найдется достаточно фантазий, чтобы составить себе таблицу побогаче, чем этот скромный образец.

Как Вы видите, размеры повреждений "привязаны" к оружию. Но оружие - это не всё, с чем Вам предстоит

встретиться в ролевых играх. Сотни самых невероятных монстров обходятся без оружия. Приличный дракон может кусаться, рвать противника когтями, бить хвостом и лапами и даже испепелять пламенем (если он огнедышащий). О тех повреждениях, которые "привязаны" не к оружию, а к монстрам, игроки узнают из специального справочника "Monstrous Compendium". Мы же в качестве примера приведём тролля, который наносит повреждение 2d6 когда кусается и 1d4+4, когда бьёт когтистой лапой, причем оба удара наносит вместе в одном раунде. Авторы компьютерных ролевых игр сами создают себе в структуре данных таблицы по монстрам, аналогичные вышеупомянутому "компендиуму".

8.7.2. Другие повреждения.

Получить по голове от какого-то тролля - не единственный способ провести время в RPG. Обычно герои неутомимо ищут и другие способы поразвлечься. И здесь у них есть дополнительные возможности.

Падение с высоты - нормальная ситуация в игре. Провалы в какие-нибудь ловушки происходят сплошь и рядом. Обычно само падение проходит нормально, хуже обстоит дело с приземлением. Принято отнимать от HP по 1d6 за каждые три метра высоты, но не более чем 20d6 (этого должно хватить, чтобы надолго успокоиться).

Паралич. Это результат специального магического воздействия. При параличе персонаж только теряет способность двигаться (и, естественно, сражаться), но продолжает дышать, видеть, думать. Если он маг, то в состоянии применять магию.

Отравление ядами. В зависимости от того, что это за яд, возможны разные степени поражения. Например, персонаж теряет половину всех своих способностей в течение 1d3 дней.

8.7.3. Лечение.

Раненого или больного персонажа можно излечивать. Излечение бывает естественное и магическое. В результате лечения он может полностью восстановить свой параметр HP, но сколько бы Вы его ни лечили, он не превысит исходное значение этого параметра до тех пор, пока не повысит свой уровень, а это достигается, как мы уже говорили, только с приобретением опыта.

При естественном лечении герои восстанавливают по 1 единице HP за каждый день отдыха. Правда, не надо понимать день отдыха, как день, проведенный в постели. Это может быть день обычного неспешного путешествия, возможно верхом, а возможно пешком, но без особых перегрузок и, конечно, без драк. Если же герои могут позволить себе полный отдых (день в постели), то темп восстановления 3 HP/день. Но и в том и в другом случае предполагается, что никаких ограничений по еде или питью у героев нет.

Магическое лечение выполняется с помощью заклинаний или с помощью магических предметов (эликсиров, волшебных палочек и т.п.). Сколько HP восстановит то или иное заклинание зависит, во-первых, от силы самого заклинания, а во-вторых, от личных способностей того, кто его произносит. При этом важно иметь в виду только одно - восстановить HP можно только до исходного уровня и не выше.

8.7.4. Летальный исход.

Когда персонаж достигает HP=0, он погибает и не может делать ничего, если не применить к нему особо мощную реанимирующую магию.

Существует понятие смерти в результате массивных повреждений. Так, если в одной атаке персонаж получает повреждение в 50 или более HP, то независимо от того, достиг он нуля или нет, он может погибнуть. В этом случае делается контрольный бросок - жить/помереть.

Существует понятие неизбежной смерти, когда HP не рассчитывается, а просто обнуляется. Например, если Ваш герой нырнет в бассейн с концентрированной кислотой, считать очки уже не надо.

8.7.5 Степени смерти.

В магическом мире все возможно и физическая смерть героя ещё не означает конца его приключений. Он может быть реанимирован, а потом излечен. Реанимация стоит денег (энергии, усилий и пр.). Но обычная физическая кончина (снижение HP до нуля) - не единственный способ умереть в ролевой игре. Есть и другие более серьёзные виды кончины. И тут все зависит от фантазии автора игры.

Например, можно представить себе ситуацию, когда параметр HP становится меньше нуля и интерпретировать её как некий другой вид смерти. В компьютерных играх нам встречались, например, "окаменение" (HP=-100) и "распыление" (HP=-1000).

8.7.6. Реанимация.

Не хотелось бы останавливаться на грустном и желательно бы здесь же рассказать и о способах реанимации и восстановления умерших героев, но, увы, естественным образом это не происходит, а для разговора о магии ещё не пришло время. Пока же скажем только, что ничего необратимого нет и, если DM запланировал оживление, персонажа, а программисты заложили это в программу, значит, игра продолжается и всё ещё впереди.

Продолжение следует…

Структуры данных и алгоритмы в RPG

© Сергей Симонович, 1994.

Цель этого цикла статей - дать основные понятия и представления для тех, кто хочет сам "спроектировать" ролевую игру, обратите внимание на то, что мы говорим "спроектировать", а не "запрограммировать". Причина этого ясна, - если Вы не умеете программировать ни на каких языках, мы Вас в данной статье этому научить не сможем. Для этого существует много других книжек, как выпущенных нами, так и не нами. Предположим всё же, что Вы умеете программировать и понимаете, что такое переменная, массив, флаг и т.п. наша задача помочь Вам одним взглядом

охватить все проблемы и задачи, которые Вам придется решать, если Вы задумаете сами написать ролевую игру.

Итак, сразу беря быка за рога, скажем, что спроектировать игру означает:

- разработать систему управления игрой;
- разработать структуры данных;
- разработать (или подобрать из готовых) алгоритмы.

Мы же начнём со структур данных.

1. Структуры данных в ролевой игре.

Если описать структуру ролевых игр двумя словами, то это обычная БАЗА ДАННЫХ. Причем неплохо было бы, чтобы у Вас эта база данных была бы графической базой данных. Так что, если у Вас есть опыт создания баз данных, то можете приступать к проектированию ролевой игры уже сейчас.

Правда, это не самая простая база данных, поскольку придётся хранить много разнородной информации и уметь ею управлять. Считайте, что это как бы связанный набор нескольких баз данных и несколько систем управления ими.

Давайте посмотрим в самых общих чертах, какие базы данных нам нужны.

1. **Географическая** база данных - в ней хранится информация о географии игрового пространства и о его структуре.

2. **Персональная** база данных - в ней для каждого героя хранится его персональное досье.

3. **База ресурсов** - в ней хранится информация о ресурсах, которыми располагает Ваш отряд в целом.

4. **Инвентарная база данных** - здесь хранится информация о тех объектах, которыми располагают Ваши герои. Оружие и доспехи не целесообразно вносить в список инвентаря.

5. **Оружейная база данных** - сведения об оружии и доспехах, имеющихся на руках у героев.

6. **База энкантеров**. Энкантерами (enconters) называются все объекты (субъекты), которые нарушают нормальную последовательность игры. Наверное, необходимы примеры. Если Ваш отряд оказывается перед запертой дверью или сталкивается где-то с монстром или находит сундук, ключ, перстень, шлем, родник, хижину, шалаш и пр. и пр. – всё это нарушает нормальную последовательность блужданий по игровому пространству. Все это энкантеры.

7. **Магическая база данных** - здесь хранится информация обо всём, что касается магии. Сколько и каких заклинаний доступны каждому члену группы и т.д. и т.п.

Вот, пожалуй и всё. О том, как эти базы связаны друг с другом (говоря научным языком, какие между ними существуют реляционные отношения), мы поговорим, когда конкретно будем обсуждать каждую из этих баз.

1.1. Географическая база данных.

1.1.1. Размерность базы.

Это массив. Разумеется, не одномерный. Размеры этого массива определяются Вашими амбициями и объёмом оперативной памяти Вашего компьютера. Можно предположить, что для игрового пространства 100x100 локаций Вы сможете выделить 10 килобайтов, если на каждую локацию уделите по одному байту. Может быть. Вам хватит 20x20 локаций и тогда Вы обойдетесь 400-ми байтами или отдадите не по одному байту на локацию, а больше.

Для хорошей ролевой игры, конечно, этого мало (разве что только для экспериментальной). И здесь, как говорится, есть варианты. Так, например, практически во всех ролевых играх игровое пространство "режут" на зоны и подгружают их по мере необходимости. Но резать можно по-разному. Можно эти "зоны" сделать с последовательным доступом, а можно и со свободным. Если игра пишется для работы с магнитофоном, то доступ может быть только последовательным. Если же предполагается использование дисковода, то проходит и тот и другой вариант. Рассмотрим примеры.

Последовательный доступ. В известной игре HEROQUEST вся игра оформлена в виде набора отдельных миссий. Каждая миссия проходит на своем игровом поле, общим (переходящим из миссии в миссию) является только набор героев и связанные с ними параметры и ресурсы, а вот географию пространства можно подгружать после исполнения каждой очередной миссии. Такая система годится для магнитофона.

Другой приём разработан в сериале EVE OF BEHOLDER. Там игровое пространство квазитрёхмерное. Герои очищают от нечисти уровень за уровнем, этаж за этажом. Им не положено с пятого этажа перепрыгивать на седьмой, пока не пройден шестой. Каждый этаж - двумерен, а последовательность этажей обеспечивает иллюзию трехмерного пространства. На первый взгляд, эта система похожа на набор отдельных миссий и здесь тоже можно было бы применить магнитофон, но это только на первый взгляд. Дело в том, что хоть Вы и не можете перейти с пятого этажа на седьмой, зато можете вернуться на четвертый, третий и т.д. Более того, иногда по ходу игры это делать просто необходимо. И вот здесь-то принцип последовательных подгрузок может подвести, причём сразу дважды. Во-первых, очень противно гонять ленту магнитофона в поисках какого-то уже пройденного уровня. А во-вторых, вместе с географией может подгрузиться и таблица энкантеров этого уровня. Представляете, как нехорошо будет вернуться на уже пройденный уровень и вновь столкнуться там с той же нечистью, которую Вы уже победили и вновь собирать там ключи и сокровища. Это был бы "прокол".

Так что либо Вы запретите своим героям возвращаться на уже пройденные этажи (например завалив дверь камнями), либо таблицу энкантеров загрузите сразу всю, без подгрузок при переходе с этажа на этаж, либо применяйте дисковод.

Свободный доступ реализован в сериале Might & Magic и в последних играх серии Ultima. И здесь есть три типа "подгрузки" новой географической базы. Во-первых, герои могут выйти за любую границу (из четырех) своей прямоугольной области. Как на шахматной доске с поля B2 можно пойти на A2, B3, B1 и C2. Возможно, что Вы разрешите и диагональные направления - на A1, A3, C1, C3. Во-вторых, в игровом пространстве немало всяких

замков, городов, подвалов, пещер и т.п. Всякая дверь, замковая решетка, ворота, вход, в пещеру - это энкантер, по которому подгружается новая географическая область, например, подземный лабиринт. И, в-третьих, в игре существует много разных способов телепортации как магической, так и научно-фантастической, забрасывающих Вас в новые области напрямую. Здесь дисковод абсолют но необходим.

1.1.2. Структура локации.

Теперь несколько слов о структуре данных по каждой локации. Если не роскошествовать и отдать, по одному байту на локацию, то уже многое можно сделать. Вот, например, что можно сделать в одном байте, израсходовав три младших бита (0…2)

0 - дорога, проход свободен, скорость максимальна.
1 - равнина, проход свободен, скорость средняя.
2 - лес, проход только для тех, кто имеет навык ориентирования.
3 - горы - проход только для тех, кто освоил альпинизм.
4 - болото - проход свободен, но велика потеря сил, есть риск утонуть.
5 - мелководье - проход для тех. Кто умеет плавать.
6 - море - проход только для освоивших навигацию.
7 - глухая стена, прохода нет и не будет.

У нас в запасе остались 5 старших битов. Третий бит целесообразно отдать под флаг, сообщающий о том, что данная локация нами изучена. Когда встанет вопрос о рисовании карты, то Вы вспомните эту предусмотрительность добрым словом. Прочие же флаги можно использовать для того, чтобы дать знать о наличии энкантера.

Бит 3 - сигнализирует о том, что эта локация уже посещалась.
Бит 4 - сигнализирует о наличии динамических объектов.
Бит 5 - сигнализирует о наличии статических объектов.
Бит 6 - сигнализирует о закрытом проходе.
Бит 7 - сигнализирует об открытом проходе.

Можно, конечно, и не отдавать биты под флаги энкантеров, а в каждой локации "лазить" в базу энкантеров и смотреть, что там, но, во-первых, это расход времени, а во-вторых, база энкантеров хоть и меньше, чем географическая база, но её нельзя "резать" (мы об этом вспомним, когда будем обсуждать вопросы сохранения отложенного состояния) и потому она оказывается довольно сложной и неудобной. Без нужды лучше к ней не обращаться. Ну и, в-третьих, стоит ведь подумать и о графической поддержке игры.

Предполагается следующий алгоритм. Встретив включенный бит-4, система "смотрит" в таблице энкантеров, что за объект (субъект) Вам попался и рисует его на экране (если он до сих пор ещё не уничтожен или не взят). Бит-5 относится к статическим объектам (например, это фонтан). Его взять нельзя, он будет рисоваться всегда. Биты-6 и 7 тоже важны. Согласитесь, по-разному выглядят закрытые проходы, если они заколочены досками, заложены кирпичей, заделаны решеткой и т.п. Так же по-разному выглядят они и когда открыты, что именно здесь у Вас - кирпич, земля, штукатурка, скала и т.д. программа узнает, когда обратится в базу энкантеров.

1.1.3. Графическая поддержка.

Это, конечно, важная самостоятельная задача, но и она решается достаточно просто и эффективно. Если отдать верхние 16 строк экрана для иллюстрации, то в них можно используя случайную графику (см. статью "Случайная графика в "ZX-РЕВЮ" № 7-8 за 1992 г., с. 161) изображать содержимое данной локации и ещё одной-двух смежных локаций. Расходы памяти при этом будут минимальны и никак не связаны с размерами Вашего игрового пространства.

Рисование должно проходить в несколько приёмов. Сначала рисуется фон, соответствующий типу местности (лес, болото, горы), а потом на него накладывается изображение энкантера. Целесообразно сразу для себя решить, что мелкие динамические объекты, такие как ключи, деньги, драгоценности, оружие и т.п. на экране изображать не стоит, а лучше об их обнаружении сообщить в строке сообщений.

1.1.4. Программные переменные.

Для работы с географической базой, к счастью, надо совсем немного программных переменных.

NUMGEO - 1 байт. Переменная указывает на номер Вашей географической зоны. Поскольку маловероятно, что у Вас будет более 255 зон, то для неё достаточно одного байта.

COORD - 2 байта. Указывает на координаты Вашего отряда в загруженной области.

DIRECT - 1 байт. Указывает на направление, в котором смотрит Ваш отряд. Это необходимо для правильного изображения графики.

1.1.5. Структура экрана.

С географией игры связан основной экран, входящий в систему управления игрой, но здесь нам надо подумать ещё о крупномасштабной карте. Поскольку RPG имеют огромный размер игрового пространства, то для работы с ними очень и очень желательно иметь возможность автоматического картографирования. Такая игра сразу многое выиграет в глазах пользователей. Но эта простая, казалось бы, функция ведет к огромным сложностям, когда мы вспомним о необходимости сохранения состояния.

Когда мы ходим по подгруженной локальной зоне, то третий бит позволит выявить уже изученные локации и показать их на карте. Но как быть, когда мы выйдем из этой локальной зоны и перейдем в следующую, а потом опять вернемся в эту же. Либо нам надо держать на диске копию зоны с включенным третьим битом, либо при повторном входе в зону программа будет "забывать", какие локации мы посещали, а какие - нет.

Оставлять на диске копию одной зоны и работать с ней динамически - это то же самое, что оставлять на диске

копии всех зон, то есть в случае, если Вы захотите прервать игру и отгрузиться, Вам пришлось бы отгружать всю географическую базу полностью, это расточительно и немудро. Конечно, если бы Вы работали на 486-ой IBM с "винчестером" в 200 мегабайтов, то могли бы позволить себе иметь отгруженный файл размером в пару Мб. Но на скромном диске TR-DOS надо поступать рачительнее.

Мы предлагаем заниматься картографированием только в пределах своей "текущей" зоны и только её отгружать при сохранении состояния. Все локации в других зонах должны быть "забыты". Таким образом, отпадает необходимость в наличии специального экрана для показа глобальной карты.

Но все-таки неразумно полностью лишать пользователя картографирования, поэтому локальную карту в пределах нескольких локаций от той точки, где он находится, все-таки дать желательно. И пусть она будет небольшой.

На рис.1 мы привели примерный вид "стандартной" раскладки основного экрана, прочно утвердившийся в RPG в последние годы.

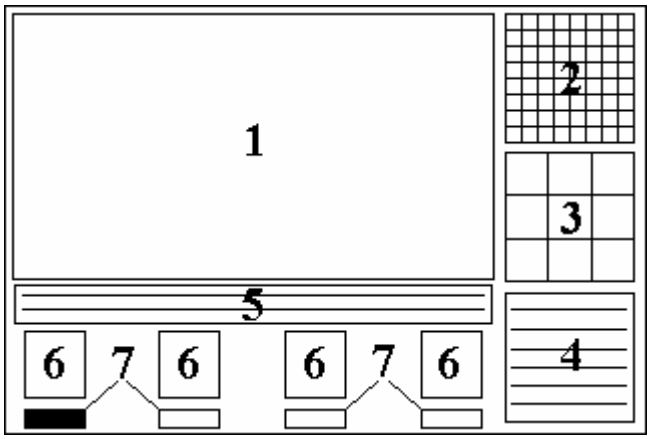


Рис. 1

1. Графическое изображение локации. Здесь изображается то, что "видит" Ваш отряд в зависимости от того, где он находится.

2. Локальная карта. Здесь с помощью символов псевдографики показана местность, окружающая Ваш отряд. Те локации, в которых Вы ещё не были, остаются закрыты ми (рис.2).

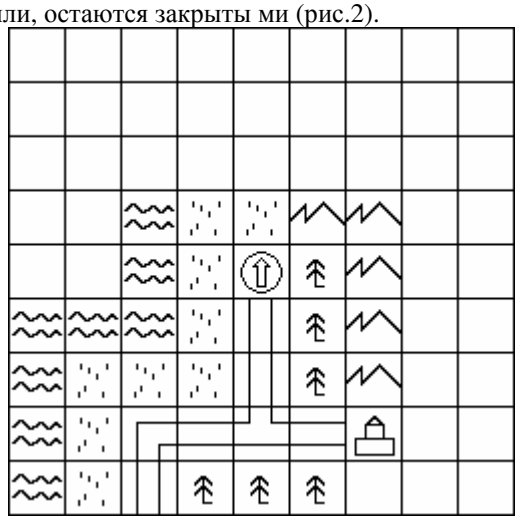


Рис. 2

Решите сами, может быть Вы будете открывать играющему не только ту локацию, где он находится но и смежные с ней.

3. Меню системы управления игрой. К нему мы вернемся в одном из следующих выпусков. Здесь Вы делаете выбор:

- куда идти;
- отдыхать;
- перейти в иные режимы игры (работа с магией, работа с инвентарем, применение метательного оружия и т.п.);

4. Групповое информационное окно. В нем дается статусная информация по отряду:

- время;
- деньги;
- еда;
- и пр.

5. Окно программных сообщений. Например, здесь Вам напишут, что "Вы утонули в болоте" или, что "Вы нашли кошелек с золотыми монетами".

6. Меню выбора персонажа. Здесь Вы можете переключаться с одного персонажа на другого.

7. Персональные информационные окна. Здесь по каждому персонажу отражается состояние его здоровья. Проще всего это делать цветом. Например, "зелёный" персонаж здоров полностью. Его параметр HP>90% от максимального. "Желтый" цвет сигнализирует о лёгком ранении - HP>70% от максимального. "Красный" - тяжёлые раны (HP>40% от максимального). "Синий" цвет сигнализирует о критическом положении (0%<HP<40%). И, наконец, чёрный цвет означает, что герой мертв и Вам недоступен.

* * *

На этом мы сегодня с Вами прощаемся до следующего выпуска, в котором начнем разговор о других базах, входящих в структуру ролевой игры.

Продолжение следует…

Описание составлено для 128K версии игры, но почти все нижесказанное относится и к 48-й версии, которая отличается от 128-й только в мелочах (отсутствуют библиотеки с предметами которые можно встретить в них, OUTPOST'ы, RAM-DISK и звук для музыкального сопроцессора).

Уже при подготовке к печати экспертной проработки Михаила Кретова, мы получили письмо Артёма Елисеева из Тулы, тоже посвященное этой игре. Оно было менее подробное, однако содержало некоторые сведения, которыми мы дополнили статью первого эксперта.

Эксперт: Михаил Кретов, г. Златоуст, Челябинской обл.

Дополнения: Артем Елисеев, г. Тула.

1. Предыстория.

История начинается в 2047 г. с изобретения "Межзвёздного привода" - принципиально нового класса двигателей, использующих для работы искусственное искривление пространства. Использование таких двигателей дало человечеству возможность развивать скорости, ограниченные лишь мощностью используемых источников энергии. Источники колоссальной мощности вскоре были созданы. Используя энергию связи кварков, выделяющуюся при расщеплении микрочастиц, они могли обеспечить скорости, превосходящие скорость света в миллионы раз. Единственной преградой, стоящей на пути будущих покорителей космоса, явилось то, что всё необходимое для работы "Межзвездного привода" топливо невозможно было взять с собой - оно взорвалось бы в момент запуска корабля. Нужно было, имея на борту сравнительно небольшой резерв этого взрывоопасного вещества, постоянно его регенерировать. Такая технология не замедлила появится - в 2049 г. группа молодых инженеров, опровергнув попутно несколько фундаментальных законов физики, предложила конструкцию так называемых "Водородных ловушек" устройств, синтезирующих топливо из рассеянных в межзвёздном пространстве веществ (в основном водорода). Перед человечеством открылись необъятные возможности по исследованию и колонизации близлежащих звёздных систем. Открылась новая эра в истории развития человеческой цивилизации.

В 2050 г. первая волна колонистов покинула Землю для высадки на четырёх ближайших звездах G-типа: Альфа Центавра, Тау Кита, Бета Гидры и Эпсилон Эридана. Семь месяцев спустя колонисты прибыли на Тау Кита и начали строительство цивилизации на третьей планете системы, в пустынном негостеприимном мире с очень быстротечным ритмом. В течение почти сотни лет колония бурно росла и процветала, быстро догоняя планету-прародительницу Землю в техническом развитии. Было построено более 30 городов и на основе сильной промышленной базы была развита робототехника, гидропоника и другие отрасли.

Но вот настал 2150 год и с ним на эту и ещё одну из четырех населенных колоний пришли эпидемии неизвестных ранее болезней. Свирепая чума и синдром Енцке унесли в течение нескольких месяцев сотни тысяч жизней. Вакцины против этих болезней вовремя не было создано, несмотря на колоссальные усилия учёных всех населенных миров. В случаях заражения, фатальный исход был обеспечен и эпидемии распространялись из одного города в другой. Немногих, оставшихся в живых пришлось срочно эвакуировать, а общение между колониями - временно прекратить. На планете остались только ремонтные роботы, которые должны были поддерживать все системы в рабочем состоянии.

Только через два года на Земле нашли вакцину, способную вернуть к жизни погибшие колонии, и связь между двумя оставшимися колониями Альфа Центавра и Бета Гидры была возобновлена.

Примерно в это время системы, слежения, наблюдавшие за оставшимися на Тау Кита автоматическими комплексами, сообщили, что над планетой взорвался огромный метеорит, в результате чего на некоторое время слежение стало невозможным. Когда пыль, поднятая взрывом, улеглась и стала возможной навигация в пределах планеты, было решено восстановить цивилизацию на поражённой планете. Уцелевшие колонисты захотели вернуться на свои родные колонии Тау Кита и звёзды Ван Маанена.

В 2164 г. первая экспедиция отправилась на Тау Кита. Корабль без неожиданностей пересек звёздную систему и благополучно приземлился на поверхность планеты. Пришло сообщение, подтверждающее удачное приземление, минутой позже был получен сигнал бедствия… а затем в эфире воцарилась мёртвая тишина. Немедленно были приведены в действие все системы поиска, но все, что они могли найти - оседающее облако пыли на том самом месте, где несколько минут назад высилась громада космического крейсера. Позже было установлено, что крейсер был уничтожен одновременным огнем полусотни боевых роботов.

С помощью дистанционно управляемых кораблей-разведчиков исследователи узнали, что взрыв метеорита привел, в частности, к разрушению систем идентификации приближающихся объектов. После взрыва метеорита,

стратегическая оборонительная система планеты пыталась защититься от новых разрушений, но, так как идентификационные системы были выведены из строя, она оказалась агрессивно настроенной по отношению к левому объекту, появляющемуся в зоне видимости радаров дальнего обзора.

В течение 7 лет собиралась информация по "взбунтовавшейся" колонии, 7 лет тщательно взвешивались все предлагаемые варианты нейтрализации вышедшей из-под контроля стратегической оборонительной системы, и, наконец, был выбран наиболее подходящий. Чтобы избежать уничтожения оставшихся на планете строений и коммуникаций, было решено набрать команду добровольцев, и, соответственно их, обучив, поручить им остановку главного термоядерного реактора, установленного в столице планеты - городе Централис. Чтобы свести к минимуму вероятность обнаружения диверсионного корабля локационными системами роботов, решили выбрать из обученных добровольцев одного, наиболее подготовленного и, снабдив его низколетящим скиммером, отправить на планету в одноместном корабле, невидимом для локаторов дальнего обзора из-за своих "игрушечных" размеров.

2. Цель игры.

Задача, стоящая перед пилотом, заключается в следующем:

1. Собрать 40 секций охлаждающих стержней, для чего придётся облететь все города планеты. Секции можно найти в реакторах небольшой мощности, которые есть в каждом городе. Кроме основного, в большинстве городов стоят резервные реакторы, в которых тоже можно найти необходимые секции. Чтобы добраться до такого реактора, нужно воспользоваться дополнительным гиперприводом, но об этом позже.

2. Смонтировать из собранных секций 20 охлаждающих стержней (каждая секция является половиной охлаждающего стержня, но для каждой половинки имеется только одна ответная часть).

3. Вставить смонтированные стержни в ячейки котла главного реактора. Каждый вставленный стержень снижает мощность реактора на 5%, то есть вставив последний стержень, Вы перекроете источник энергии роботов, и, смертоносная военная система замрет, ожидая, пока её демонтируют.

3. Боевой скиммер.

GCS-скиммер - мощная и опасная боевая машина, сплав самых передовых технологий и новейших инженерных решений, лучшее, что было создано человечеством в области вооружений за всю его историю. И Вам предстоит управлять этим чудом современной техники.

Скиммер укомплектован следующим вооружением:

- рубиновый лазер повышенной мощности с системой охлаждения (LASERS);
- защитный энергетический экран сферической формы (SHIELD);
- 8 реактивных ракет с головками теплового самонаведения (MISSILES);
- многофункциональная система сопровождения целей;
- 8 антиракетных снарядов (AMM'S);
- 8 осветительных ракет (FLARES);
- система инфракрасного видения (у планеты нет спутника, подобного Луне, поэтому после заката можно видеть с помощью этой системы);
- круговой радар непрерывной индикации (SCANER);
- 4 переключаемых обзорных эк рана (FORWARD-, RIGHT-, REAR-, LEFT-VIEW);
- гироскопический компас (COMPASS) и указатель направления на базу (ADF);
- бортовой компьютер со специализированной базой данных.

Управление скиммером возможно в двух режимах: РЕЖИМЕ ПОЛЁТА и СТАТИЧЕСКОМ РЕЖИМЕ. Первый, в общем-то, является основным, поскольку именно в нем возможно исследование планеты и поиск необходимых предметов. В этом режиме управление осуществляется с клавиатуры или джойстика (можно играть и KEMPSTON-джойстиком, если в режиме переопределения клавиш двигать джойстик в соответствующую сторону). Управляющие клавиши можно просмотреть и переназначить с помощью команды KEYS статического режима (см. ниже). Стандартным является следующий набор клавиш:

O - влево.
P - вправо.
S - ускорение/курсор вверх.
X - торможение/курсор вниз.
SPACE - огонь лазером/окончание выбора.
M - огонь ракетами.
F - огонь осветительными ракетами.
A - огонь противоракетными снарядами.
R - вывод на экран текущего состояния игры.
I - включение/выключение системы инфракрасного видения.
H - увеличение высоты.
G - уменьшение высоты.
J - активизация системы телепортации.
L - приземление и переход в статический режим (при нулевой высоте).
V - переключение обзорных экранов.

В статическом режиме управление скиммером осуществляется посредством прямого диалога с бортовым компьютером. Список допустимых команд выводится на экран командой HELP. Назначение команд:

HELP - вывод перечня допустимых команд.

LAUNCH - перевод скиммера из статического режима в режим полёта.

MAP - вывод на экран карты планеты. С помощью символов управления Вы можете двигаться по карте в любом направлении и изменять масштаб карты.

RODS - включает манипулятор для работы с охлаждающими стержнями. Чтобы смонтировать стержень из двух подобранных секций (при наложении друг на друга они должны составлять полный круг с черной полосой посредине), нужно, изменив состояние (цвет, горизонтальное и вертикальное расположение) одной секции в соответствии с состоянием другой, наложить их друг на друга. Если Вы все сделали правильно, то компьютер подтвердит удачное окончание операции и сообщит количество собранных стержней. В противном случае Вы увидите сообщение о нарушении работы системы.

REACTOR - дает доступ к управлению главным реактором. Естественно, работает только в том случае, если Вы к нему пристыкованы. Внимание! Не активизируйте систему управления главным реактором, не имея на борту защитного скафандра - без него Вы быстро погибнете из-за сильной радиации и высокой температуры. Даже в скафандре Вам придется торопиться, чтобы успеть вставить все 20 охлаждающих стержней. Вставлять их надо в те ячейки котла, которые в данный момент являются активными. Вставив, последний стержень, Вы выполните свою миссию и с почётом возвратитесь на Родину.

LOOK - по этой команде бортовой компьютер сообщит Вам, где Вы находитесь в данный момент и, если Вы находитесь в каком-либо здании, покажет вид этого здания изнутри.

WAIT - переводит бортовой компьютер в режим ожидания на 5 минут (земных, или около 1/12 местных суток).

STATUS - сообщает о повреждениях и показывает счёт (то же, что клавиша R в режиме полета).

EQUIP - позволяет отремонтировать, дозаправить Ваш скиммер и пополнить его вооружение.

NAME - сообщает компьютеру имя нового пилота.

PAD - вызов бортового журнала. Предоставляет в Ваше распоряжение небольшой текстовый редактор, с помощью которого можно записать то, о чем Вы не хотели бы забыть.

NEWPAD - очищает бортовой журнал.

SAVE - запись состояния игры на магнитный носитель.

LOAD - загрузка состояния игры

RAMSAVE - быстрая запись состояния в оперативную память. Эта информация теряется при выключении или перезапуске компьютера.

RAMLOAD - загрузка состояния из оперативной памяти, если ранее туда была произведена выгрузка.

KEYS - позволяет посмотреть и переназначить управляющие клавиши для режима полета.

QUIT - перезапуск игры.

SCORE - Ваши результаты.

PAUSE - приостановка игры до нажатия FIRE.

SIGHTS ON / SIGHTS OFF - включение/выключение прицела.

LIBRARY - дает доступ к городской библиотеке, если Вы к ней пристыкованы.

TIMEVAULT - активизирует систему открытия кодового замка найденного в библиотеке тайника.

INVENTORY - выдает список объектов, взятых из тайника на борт.

Все перечисленные команды можно сокращать, добавляя к сокращению знак "." (точка). Так, "H." будет понято как "HELP", "L." как "LAUNCH", а "LIB." - как "LIBRARY. Кстати, если набрать просто ".", то компьютер также поймет это как "HELP". Для опознавания команд Ваш бортовой компьютер использует так называемый "принцип неопределенных сравнений", что позволяет ему не обращать внимания на ошибки в написании командных слов.

4. Навигация.

Планета оборудована стандартной телепортационной системой, облегчающей перемещения из одного города в другой. К несчастью, действуют телепорты только в одном, определенном направлении. Но бортовой компьютер Вашего скиммера оснащен полной картой гиперприводов телепортов. Чтобы попасть в нужный вам город, найдите соответствующую JUMP-платформу и, находясь в непосредственной близости от нее, нажмите "J". Соблюдайте крайнюю осторожность при выходе из гиперперехода, чтобы не столкнуться с "Песчаным прыгуном" - местной формой жизни, обитающей вокруг JUMP-платформ. Кроме того, Вас могут поджидать "Гюнтеры" - роботы-охотники, так что будьте готовы отразить атаку.

В отличие от всех остальных объектов, Ваш лазер не может причинить JUMP-платформе никакого вреда, но Вы можете разбиться о столб, отмечающий её местоположение.

Несколько слов о топографии городов. По карте планеты Вы можете определить их Относительное расположение, которое пригодится Вам для совершения гиперпереходов. Чтобы попасть в город, находящийся к северу от того, в котором Вы находитесь, найдите JUMP-платформу, расположенную у северной границы вашего города - она и будет той, которая Вам нужна. Теперь нажмите "J." - и окажетесь в южной части нужного Вам города. Кроме JUMP-платформ в городах Вам встречаются различные сооружения.

БИЛИОТЕКА (LIBRARY). Является хранилищем самой разнообразной информации. Библиотеки имеют огромный банк данных, в котором можно найти сведения об основателях колонии, оружии, звездах, соседних колониях и т.д. также в библиотеках можно получить данные на абсолютно все наземные объекты, начиная с роботов-охотников, строений и кончая местными формами жизни, о которых предупреждает Галактическая Корпорация. В библиотеке можно посмотреть карту города и его OUTPOST'а.

Дав команду LIBRAARY, Вы окажетесь в главном меню. Здесь Вы можете выбрать:

ACCESS DATA BANK - доступ к банку данных. Введите непонятное слово или словосочетание, и, если оно есть в банке данных, компьютер объяснит Вам его значение. В общем, это небольшой электронный толковый словарь.

CITY MAPS - карты городов. Вы можете просмотреть Карты Вашего города и ближайших к нему городов, а также карту "сторожевой заставы" (OUTPOST) Вашего города.

IDENTIFICATION CHART - идентификационная таблица. Советую внимательно её изучить, прежде чем приступать к выполнению миссии. В разделе "СТРАТЕГИЯ" будут перечислены Ваши враги. Посмотрите и запомните, как они выглядят - это поможет Вам не дать застать себя врасплох.

EXIT LIBRARY UNIT - выход из библиотеки.

Иногда в библиотеке оказывается тайник, оснащенный кодовым замком. Открыв его, можно обнаружить в нем различные полезные предметы, которые можно будет использовать в дальнейшей игре. Об этом Вы узнаете по сообщению бортового компьютера "TIMEVAULTS FOUND IN LIBRARY". В этом случае дайте команду TIMEVAULT и приступайте к "взлому" кодового замка Тайника, который заключается в следующем. Включите генератор кода, выбрав пиктограмму с изображением стрелок. Вокруг этой пиктограммы расположены в виде восьмигранника 8 индикаторов, на которых после включения генерации установится определенная картинка. Одновременно в левой части экрана начнется отсчет времени. Выберите любой из 8 индикаторов. Если при этом после звукового сигнала изменила цвет первая точка в линейке, расположенной вверху экрана, и снова началась генерация кода, значит Вы угадали первый из 8 элементов кода. Теперь Вы увидите ту же картинку, но несколько повернутую вокруг оси. Вам остается только выбрать соответствующий индикатор и повторить те же операций ещё 6 раз. Когда весь код будет подобран и тайник отпёрт, перед Вами появляется отлично нарисованная картинка, изображающая его содержимое. Компьютер сообщит, что это такое и напишет, как этим пользоваться. Если же Вы ошиблись при наборе кода, отсчёт времени будет остановлен, и Вам придется начинать всё сначала.

Теперь рассмотрим, какие предметы вы можете найти в тайнике.

EXPERIMENTAL MISSILE - экспериментальная ракета. Очень полезное приобретение - уничтожает все предметы (кроме имеющих уровень защиты VERY HIGH) в пределах досягаемости Вашего радара. Активизация - SS+M.

RESERVE SHIELD - резервный блок энергетической защиты. Включается автоматически при снижении уровня основной защиты ниже критической отметки. Но эффект от его включения мал, поскольку к этому моменту обычно повреждается большинство оборудования в том числе и компас.

SINGLE JUMP UNIT - автономная телепортационная установка. Служит для перемещения из любой точки города в соответствующую точку "сторожевой заставы" (OUTPOST), и обратно. Включение – клавиши SS+J.

DURIUM CRYSTAL - предмет, придуманный, очевидно, для того, чтобы усложнить нам с Вами жизнь. Дело в том, что до некоторого времени он себя никак не проявляет, пока в один прекрасный момент Вы не вылетаете из какого-либо здания. Вас встряхивает, как если бы Вы налетели на мину, и Вы обнаруживаете, что все системы, которые могли выйти из строя, вышли из него, а энергии остается - кот наплакал. Кроме того, из вооружений у Вас остаются только осветительные ракеты, так что, если в пределах видимости нет центра поддержки, шансов выжить у Вас нет никаких. Однако правила в компьютерных играх придуманы как раз для того, чтобы их обходить. Поэтому перед каждым "взломом" тайника давайте команду RAMSAVE, и, если после "взлома" это окажется DURIUM CRYSTAL, загрузите состояние через RAMLOAD и постарайтесь не трогать этот тайник до окончания игры.

Кроме всего вышеперечисленного. Вы можете найти защитный противорадиационный скафандр. Надевается он автоматически при входе в главный реактор. Теперь можно находиться в помещении главного реактора большее время.

РЕАКТОР (REACTOR). Здесь можно найти секции охлаждающих стержней. В случае если они есть, они автоматически помещаются в магазин манипулятора, из которого могут быть извлечены для различных операций после подачи команды RODS.

ГЛАВНЫЙ РЕАКТОР (MAIN REACTOR) . Кроме того, что можно делать в обычных реакторах, возможно выполнение команды REACTOR.

ЦЕНТРЫ ПОДДЕРЖКИ (РЕМОНТНЫЕ) (SUPPLY CENTRE). Встречаются два вида центров поддержки - военные (MILITARY) и гражданские (CIVILIAN). В первых Вы можете получить весь спектр услуг, необходимых для восстановления боеспособности Вашего скиммера - ремонт, дозаправка, пополнение боезапаса, в то время как в гражданских Вам не смогут дать ракеты и противоракетные снаряды.

БАЗОВЫЙ КОМПЛЕКС ГАЛАКТИЧЕСКОЙ КОРПОРАЦИИ (GAL-CORP LANDER). Отсюда Вы начинаете игру, и единственное подобное сооружение на планете расположено в городе REEMA. По возможностям экипировки аналогично военному центру поддержки.

Небольшая деталь: при стрельбе по базе шлюз не закрывается, а если сделать хоть один выстрел по библиотеке, реактору или центру поддержки, доступ внутрь данного строения будет закрыт навсегда, что отнюдь не радует.

5. Стратегия.

Кроме перечисленных сооружений (имеющих уровень защиты VERY HIGHT - они были перечислены выше), Вы столкнетесь также с другими объектами. Наибольшую опасность для Вас представляют "сторожевой скорпион" (GUARDIAN CRAWLER) и роботы-охотники (HUNTER MK I, II, III). Безумные, хорошо вооруженные "Гюнтеры" и управляющие ими кибер-гвардейцы обычно встречаются группами по три-четыре, но иногда Вас будут поджидать засады по 10-15 "Гюнтеров". Попав в поле Вашего зрения, они некоторое время выжидают, а затем быстро

разгоняются и атакуют Вас, поэтому, если Вам предстоит бой с большой группой охотников, постарайтесь делать так, чтобы в пределах досягаемости радара было одновременно не более 3-4 противников - с большим количеством Вы не справитесь. Дождитесь, пока они на Вас нападут, и уничтожьте первую группу, затем осторожно продвиньтесь дальше и дождитесь нападения следующей, И так до тех пор, пока вся армада противника не превратится в пыль.

Не менее опасны, хотя и неподвижны, тяжело вооруженные крепости (FORTRESS), башни слежения (CONTROL TOWER), "сферы" (SPHERE), "куполы" (DOME). Все они не обладают мобильностью, но открывают огонь, как только Вы приближаетесь к ним на достаточное расстояние.

Радар с орбиты обнаружил также наличие мин (PROXIMITY MINE), поэтому во время движения внимательно смотрите "под ноги" - достаточно двух-трех взрывов мин, чтобы отправить Вас к праотцам.

Достаточно часто встречается ремонтный робот (REPAIR DROID) - неагрессивное существо, никогда не стреляющее первым, но отвечающее огнем на огонь.

Все остальное, что встретится Вам в процессе исследования планеты, не представляет абсолютно никакой угрозы и совершенно не защищено даже от огня лазеров.

6. Дополнительны сведения.

Планета Тау Кита 3 имеет очень короткий день, примерно равный одному земному часу, и такой же короткий год - 16 земных или 384 местных суток. Кроме того, у этой планеты нет спутников, поэтому в тёмное время суток лучше воспользоваться системой инфракрасного видения или осветительными ракетами.

Основные строения на поверхности оснащены последней модификацией автоматических воздушных шлюзов. Их наличие можно определить по характерному мерцающему узору. Для стыковки нужно на минимальной скорости приблизиться к шлюзу и немного подождать, пока вход полностью откроется. После стыковки компьютер автоматически перейдет в статический режим управления.

7. Заключение.

Галактическая Корпорация обращается к Вам с настоятельной просьбой: старайтесь избегать бессмысленного уничтожения гражданский строений - реколонизация начнется сразу после завершения Вашей миссии, и каждое сохраненное здание сэкономит немалые средства при восстановлении цивилизации на планете.

* * *

OCEAN CONQUEROR

Наш читатель Мажников Д.В. прислал несколько дополнений к описанию этой игры, которое мы опубликовали в ZX-РЕВЮ № 2'94.

Во-первых, для грубого (быстрого) разворота перископа используются комбинации клавиш CS+1 или CS+2, и лишь для плавной подгонки прицела используются 1 и 2.

Во-вторых, перископ - гораздо более полезная вещь, чем может показаться. Возможно он имеет такую конструкцию: на крыше рубки установлена вращающаяся пусковая установка, на ней закреплен перископ высотой около 4-х метров. При плавном погружении до глубины XXXX-4, под водой скрывается рубка, пусковая установка, остается лишь вершина перископа. (XXXX - уровень моря, т.е. глубина в надводном положении.) Следует заметить, что длина воздухозаборного шланга - 5 метров, поэтому Вы можете идти (не говорите моряку, в т.ч. подводнику, плыть - обидится!) на дизеле на глубине XXXX-5, правда уже по карте, т.к. перископ скроет под водой.

На этом можно строить интересную и эффективную тактику боя. Погрузившись на глубину xxxx-4, идите на сближение с конвоем. Когда Вы уничтожите один сторожевик, второй кинется к Вам, не видя Вас, ориентируясь лишь по пенному следу перископа и не стреляя. Плавно и не спеша, прицельтесь и уничтожьте его ракетой.

Таким образом, можно подойти к конвою очень близко (вплоть до тарана лодкой сторожевика) и остаться незамеченным.

Одно НО: в пылу боя следите за рулями глубины - управляя ракетой, их легко "сдернуть", лодка очень незаметно (поверьте!) всплывёт, и Вас начнут обстреливать.

THE GREAT ESCAPE

Дополнения к описанию этой игры, которое мы опубликовали е ZX-РЕВЮ № 3'94, прислал нам Олег Балуев из г. Стрвжевой Томской обл.

Во-первых, подземный ход, о котором писал А. Школьников не ведёт на волю, он ведёт на площадку для прогулок. Для того чтобы расчистить проход, Вам понадобится лопата, но об этом позже. Вернёмся к подземным ходам - их в игре два. Первый ведет от комнаты до площадки, а второй расположен в "коридоре", попасть в который можно, если во время прогулки выйти с площадки в первые ворота и идти вниз, а затем повернуть и дойти до второй смотровой вышки. Обратите внимание на землю - здесь, можно увидеть вход, он выглядит как квадрат с точками внутри. Для того чтобы попасть в подземный ход "потопчитесь" по квадрату. Ход выведет Вас в комнату, где лежит лопата. Берите её, и бегите в первый подземный ход (тот, который за печкой). Ну вот, теперь можно расчищать проход.

Я, к сожалению, тоже не смог найти применение тем предметам, о которых пишет А. Школьников, но про шоколадку могу замолвить слово. Странный эффект получается, если выложить шоколадку на печку. Печка "затанцует", превратится в человека, потом опять в печку, опять, в человека и, наконец, встанет напротив входа в комнату, что очень помогает, если за Вами гонятся охранники. Нужно вбежать в комнату и закрыть печкой проход -

теперь через эту дверь охранники войти не смогут.

Теперь об охранниках. Все они одеты в шинели и фуражки, и ходят с ружьями наперевес. Все, кроме одного - коменданта лагеря. Он одет по иному, чем остальные солдаты, и обладает способностью различить Вас даже тогда, когда на Вас надета форма.

Ну, а теперь о самом главном - о предметах. Оказывается, что не все предметы открыто лежат на полу или земле - некоторые спрятаны, и, причём неплохо. Так, например, под одной из наблюдательные вышек можно найти крайне полезный предмет - ещё один ключ. Этим ключом можно открыть самую "нижнюю" дверь (но не ту, которая в коридоре из заборов). Войдя в эту дверь, можно попасть в коридор, в котором есть ещё две двери. Одна из них открыта. Войдите в неё, и Вы окажетесь в комнате, где лежат очень-очень полезные отмычки. Они открывают любую дверь, но следует учесть, что для того, чтобы открыть дверь отмычками потребуется время, чтобы подобрать ту, которая нужна. За одной из дверей здания, в которой размещается столовая, лежат ещё два предмета: это третий ключ и непонятный предмет странной формы, который очень напоминает лист металла. Применить их нигде не удалось.

Теперь о дверях. Если у Вас есть отмычки, то двери для Вас не преграда. Куда сложнее играть, если у Вас кроме ключей ничего нет. В этом случае большинство комнат здания будут для Вас закрыты. Однако в игре есть дверь, которую не удалось открыть даже отмычками. Это самая "нижняя" дверь, которая расположена между двумя заборами. Пробраться к этой двери крайне сложно, если у Вас нет шинели, т.к. в узком пространстве между заборами несут вахту не только люди, но и собаки. Я не знаю, каким образом можно открыть эту дверь. Может быть, с помощью третьего ключа, а может как-то ещё. Возможно, именно через эту дверь можно убежать из лагеря? Ответ только один т "Играйте, думайте, ищите!"

Надеемся, эти дополнения помогут Вам наладить лучший контакт с программами.

* * *

ЗАЯВКА: И022
АВТОРСКИЙ ПЕРЕВОД: BOULDER DASH 2+
АВТОР: Борис Круглыхин (KIDSONsoft), Новосибирск.
ДАТА ПЕРЕВОДА: 3 мая 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
ЖАНР: Ходилка-собиралка.
КРАТКОЕ ОПИСАНИЕ:

Оригинал - программа BOULDER DASH 2 (FIRST STAR SOFTWARE, 1985). В предлагаемой версии отключено время и счетчик жизней. Могут играть два человека. Управление - любое.

****

ЗАЯВКА: И023
АВТОРСКИЙ ПЕРЕВОД: NEW POKER
АВТОР: Борис Круглыхин, (KIDSONsoft), Новосибирск
ДАТА ПЕРЕВОДА: октябрь 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Откомпилированный Бейсик + Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
ЖАНР: Traditional.
КРАТКОЕ ОПИСАНИЕ:

Перевод игры POKER'85 (GEORGE). Это игра в покер для четырёх человек, имеются часы. Стартовый капитал - 1 млн. $. Если кончаются деньги, есть возможность заложить дом, яхту и т.д.

****

ЗАЯВКА: И024
АВТОРСКИЙ ПЕРЕВОД: DOMINO+
АВТОР: Борис Круглыхин (KIDSONsoft), Новосибирск.
ДАТА ПЕРЕВОДА: март 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Бейсик.
НАЛИЧИЕ ЗАЩИТЫ: не указано.
ЖАНР: Traditional.
КРАТКОЕ ОПИСАНИЕ:

Перевод игры DOMINOES (Phipps Associates 1983). В начале игры - бросается жребий, кому ходить. В процессе игры выводятся все символы домино, которые в игре. Есть встроенная инструкция.

****

ЗАЯВКА: И025
ПРОГРАММА: ZODIAK STRIP 2
АВТОР: Борис Круглыхин (KIDSONsoft), Новосибирск.
ДАТА РАЗРАБОТКИ: 9 мая 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Бейсик + Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
ЖАНР: Traditional.
КРАТКОЕ ОПИСАНИЕ:

Усовершенствование оригинальной версии ZODIAK STRIP (SUSY SOFT, 1986). Программе на русской языке. Для входа в программу нужен 4-значный пароль. Как и в оригинале можно просмотреть десять картинок, но для просмотра каждой нужен 5-значный код.

****

ЗАЯВКА: П006
ПРОГРАММА: "АДАПТЕР РИСУНКОВ"
АВТОРЫ: Николай Шмаков и Алексей Щегольков "F.O.R.C.E. Ltd", г. Усть-Илимск.
ДАТА РАЗРАБОТКИ: август 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: не указан
НАЛИЧИЕ ЗАЩИТЫ: не указано
ДОКУМЕНТАЦИЯ: встроена инструкция
КРАТКОЕ ОПИСАНИЕ:

Программа позволяет адаптировать рисунки под нестандартные загрузчики. Имеется пять вариантов адаптации типа BOMB JACK, ATHENA, CHASE H.Q. и два новых варианта загрузки. После проведения адаптации блок загрузки с рисунком отгружается на ленту.

****

ЗАЯВКА: И026
ПРОГРАММА: "ТРАКТОР"
АВТОР: Щелканов Михаил Юрьевич (Mouse), г. Братск.
ДАТА РАЗРАБОТКИ: 15 августа 1994 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер
НАЛИЧИЕ ЗАЩИТЫ: есть

ДОКУМЕНТАЦИЯ: описание игры
КРАТКОЕ ОПИСАНИЕ:

В этой программе использована идея широко известной игры "Sokoban". В игре 40 уровней, при попадании в таблицу результатов, станет, возможна запись результата на кассету.

****

Вы можете приобретать литературу "Инфоркома" по следующим адресам.

г. Москва, ул. Новый Арбат, д.2. 19-е отделение связи, 1-ый этаж операционного зала, с 10 до 17 ч., обед с 14 до 15.

г. Москва, радиорынок в Митино, проезд до ст. метро Тушинская или поездом до пл. Трикотажная (Рижское направление). Суббота, воскресение, 9-14, место А-37.

г. Москва, радиорынок в Царицыно, проезд до ст.метро Царицыно. Суббота, воскресение, 9-14.

г. Архангельск, 163051, Архангельск-51, а/я 51.

г. Белгород, магазин "РАДИОТОВАРЫ", ул. Ленина, 32.

г. Белгород, ул. Октябрьская, д.84, кв. 103, "СТУДИЯ КОМПЬЮТЕР"

г. Брянск, пр. Ленина, учебный корпус № 2 Брянского строит, техникума, фойе, ТОО "ЗЕФА".

г. Владивосток, Океанский проспект, 140, магазин "ПАНОРАМА". Проезд трамваем до ост. Некрасовская.

г. Воронеж, студия компьютерных игр SAN-SAN. Магазин-салон "ЭЛЕКТРОНИКА", тел. 14-00-73.

г. Днепропетровск, ул. Шевченко, 34. Фирма "ЭКОС".

г. Екатеринбург, магазин "СПЕКТРУМ", Главный проспект, 99.

г. Екатеринбург, ул. Коминтерна, 5, общежитие РТИ, магазин "РАДИСТ".

г. Екатеринбург, киоски "РАДИОДЕТАЛИ":

- ул. Малышева, магазин "ОРБИТА",
- ул. Белореченская, магазин "РАДИОТОВАРЫ",
- ул. Блюхера, магазин "ОКРУЖНОЙ УНИВЕРСИТЕТ".

г. Ижевск, Удмуртия. Радиорынок "БЕРЗОВАЯ РОЩА".

г. Кемерово, магазин "ТЕХНИЧЕСКАЯ КНИГА", ул. Весенняя, 24.

г. Кемерово, магазин "ОРБИТА", пр. Ленина, 133.

г. Киров, "Дом науки и техники", Магазин-салон "МАРС",ул. Производственная, д.27.

г. Красноярск, радйорынок, проезд до ост. "Затон", суббота, воскресение

г. Минск, Сторожевский радиорынок, сектор 8, место 7.

г. Набережные Челны, Татарстан, Новый Город, салон-магазин "ПРИНТЕР".

г. Нижний Новгород, ТОО "КРИОНН", магазин "ФОТОЛЮБИТЕЛЬ", ул. Горького, 146. i

г. Оренбург, магазин "ВОЕННАЯ КНИГА", ул. Советская.

г. Пермь, Комсомольский проспект, 86, магазин "ЮНЫЙ ТЕХНИК", тел. 64-32-23.

г. Прокопьевск, Кемеровской обл., ул. Артема 8, магазин фирмы "АРБАТ".

г. Рыбинск, ул. Гоголя,1. ТТЦ "ГНОМ".

г. С.-Петербург, радиорынок "АВТОВО", киоск № 49, ("SCORPION-256").

г. С.-Петербург, фирма "Скорпион", тел. 524-16-53, 248-47-69.

г. Таганрог, радиорынок, суббота, воскресение, проезд троллейбусом 1,5 до ост. "Центральный Рынок".

г. Тула, Мясковксий рынок, суббота, воскресение, фирма "ТИМКО".

г. Тула, 300012, Тула-12, а/я 409, фирма "ТИМКО".

г. Челябинск, ул. Елькина, 45, магазин напротив Публичной Библиотеки.

г. Ярославль, магазин "РАДИО-СПОРТ-ТУРИЗМ", отдел "Электроника", Ленинградский проспект.

г. Ярославль, Московский проспект, д.153, ТО магазин "ГАММА".

Вниманию дистрибуторов!

Адреса наших оптовых покупателей публикуются бесплатно.