

«ИНФОРКОМ»



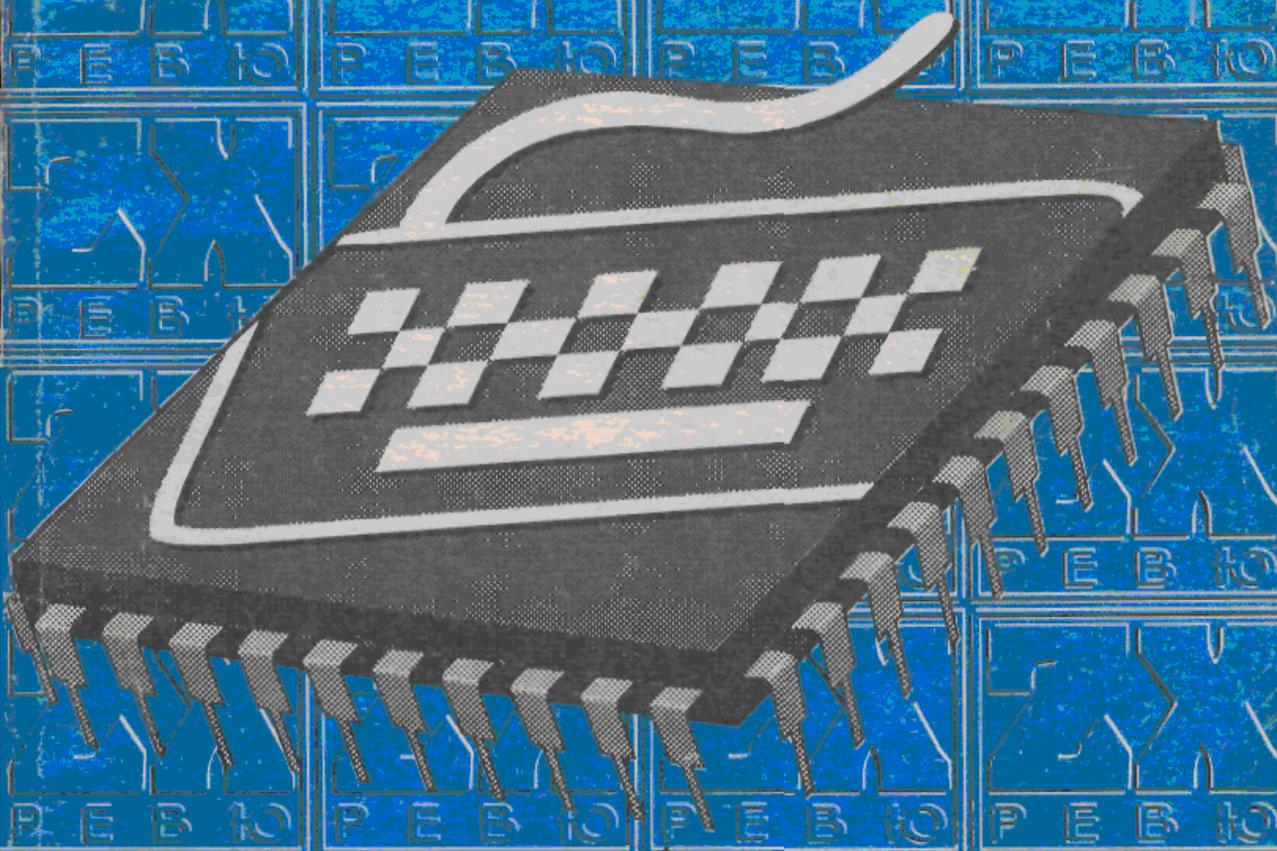
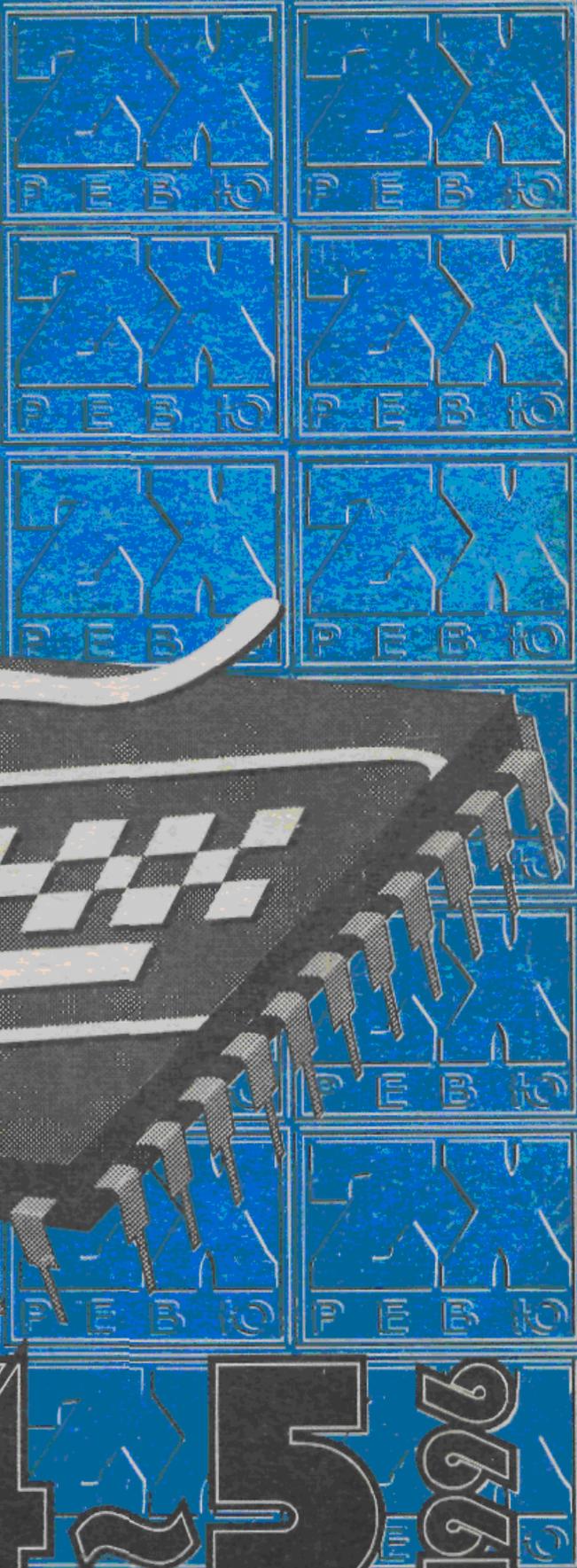
ZX-РЕВЮ
№ 4-5'96



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



P · E · B · IO



425 1996

121019,
Москва,
а/я 16,
МКП
"Инфорком"



'96
Выпуск 4-5

СОДЕРЖАНИЕ

ОТ РЕДАКЦИИ		3
АВТОРСКАЯ РАЗРАБОТКА		5
Адаптация игр 8-битовых видеоприставок для ZX Spectrum с видеопроцессором	С. Веремеенко	5
Винчестер для ZX Spectrum	С. Веремеенко	15
НОВЫЕ ПРОГРАММЫ		18
STS v4.2	Stalker	18
Теневой макроассемблер-отладчик ALASM 128/512 v2.5	А. Мищенко	20
TR DOS для начинающих		22
ЧИТАТЕЛЬ - ЧИТАТЕЛЮ		32
MULTICOLOR - эффекты на бордюре и кое-что еще	С. Кучкин	32
Получение случайных чисел	А. Астафьев	54
КОМПЬЮТЕРЫ, КОТОРЫЕ МЫ ВЫБИРАЕМ		61
Доступ к верхней области памяти	П. Шилиев	81
ЭТЮДЫ		63
О табличном программировании	А. Астафьев	63
Вывод символов в восьмикратном увеличении	И. Пивен	66
Индикатор амплитуды звука	А. Васильев	69
Бегущая строка	А. Васильев	70
Способ вывода изображения на экран	А. Васильев	72
Эффекты для музыкального сопроцессора	Д. Кравников	73
Процедуры "проявления" экрана	И. Командин	74
Процедура-календарь	А. Сидоренко	75
Процедура поиска шрифтов и спрайтов	Е. Котляров	75
Оригинальная бегущая строка	Е. Котляров	77
Процедура опроса клавиатуры	А. Шокало	78
Программа гашения экрана случайным образом	А. Волков	79
Загрузка Бейсик-программ без автостарта	А. Прокопенко	80
Компрессор символьной информации	М. Тамаринов	81
Быстрая печать шаблона 8x8	М. Тамаринов	82
ФОРУМ		83
Установка порта #FD	Ю. Васин	83
Жесткая дешифрация порта #1FFD в ПК Scorpion ZS-256	Е. Тувиков	84
Доработка STS 3.3	А. Ермишин	84
Доработка TASM 128	А. Мигунов	86
Доработка посекторных загрузчиков	А. Соловьев	86
Нестандартные магнитофонные загрузчики	М. Торгунов	88
Процедура печати 42 символов в строке	О. Литвинов	90
Настройка дисководов	С. Колотов	101
О программе G.A.C.	А. Матюшев	103
О вызове подпрограмм TR-DOS	Alex	105
Об управлении KP181BF93	С. Комиссаров	105
О компьютере "Hobbit"	П. Лебедев	105
Усовершенствованная версия Advanced Drive FX	П. Лебедев	106
А. Рубин. Недокументированные операторы LASER BASIC	П. Лебедев	107
О настройке дисководов	В. Кекух	109
Вопросы сегментации экрана	И. Пивен	110
Магнитофонный загрузчик с индикатором	Б. Артемьев	111

Об игре "LIFE"	А. Александров	112
Проблемы развития ZX Spectrum	А. Диденко	113
ФОРУМ-ИГРЫ		115
Рассмотрены следующие игры:	VENTURAMA	
THE SCEPTRE	CARRIER COMMAND	
ELITE	SIM CITY	
ROBIN OF THE WOOD	THINK	
HACKER 2	RAMPAGE	
ЗВЕЗДНОЕ НАСЛЕДИЕ	"A.T.F."	
COLONY		
ПЕРЕКРЕСТОК		134
Рассмотрены следующие ад- вентюры:	TERRORMOLINOS	
HE-MAN	SAM SPOON	
HAMPSTEAD	SPIDERMAN	
ЗВЕЗДНОЕ НАСЛЕДИЕ	KENTILLA	
REBEL PLANET	HEAVY ON THE MAGIC	
THE FAMOUS FIVE	MAGNETIC MOON	
ADVENTURE PROJECT		141
Анализ частей речи в русскоязычных ADVENTURE	К Дзреев	141
СОВЕТЫ ЭКСПЕРТОВ		147
ROCK STAR ATE MY HAMSTER	О. Бобрышев	147
DUNDEES EUROPEAN CHALLENGE	Д. Сухарев	157
WINTER GAMES	А. Рубин	158
CARRIER COMMAND	Light	160
THE GHOUSTBUSTERS 2	А. Паршин	164
HACKER	О. Бобрышев	166
LAST BATTLE	С. Хромов	170
	Е. Селиверстов	
	О. Смолянкин	175
CAPTAIN FIZZ		177
КОМПЬЮТЕРНАЯ НОВЕЛЛА		
История в момент творения (по игре MYTH)	А. Березин	177
NEVER EARTH	С. Татаринцов	183
TRICKS ZX (КОДЫ, ПАРОЛИ, ХИТРОСТИ)		188
СПРАВОЧНАЯ ИНФОРМАЦИЯ		192

ОТ РЕДАКЦИИ

Здравствуйтесь, дорогие друзья!

Вы держите в руках сдвоенный (4+5)-й номер ZX РЕВЮ. Впервые на 6-м году своего существования эти номера выходят так рано. Огромное спасибо всем соавторам нашего с Вами журнала - ЕДИНСТВЕННОГО В МИРЕ полиграфического издания по проблемам ZX-SPECTRUM.

Напоминаем всем нашим читателям, что в этом году мы выпустим три дополнительных номера (7, 8, 9) ZX РЕВЮ, на которые «объявлена дополнительная подписка (шифр КИ15-Б в нашем прайс-листе). Просим всех, кто заинтересован в увеличении объема материалов ZX РЕВЮ и других наших изданий, не откладывать подписку на дополнительные номера 1996 года и на 1997 год, Т.к. от Вашей активности на этом этапе зависят наши издательские планы на будущее.

Сейчас в портфеле редакции находятся четыре готовые к изданию книги. Это уже объявленные ранее "МИР ЗВУКОВ СПЕКТРУМА" П. Лебедева из Мурманска и "TRICKS ZX", подготовленная редакцией. Другие две - это потрясающий фундаментальный труд П. Фебина (г. Москва) "ПОЛНОЕ ОПИСАНИЕ И ПОЛНЫЙ ДИЗАССЕМБЛЕР TR DOS 5.04T (5.03)", а также третий том серии JOKER - "ПЕРЕКРЕСТОК ДРАКОНОВ" – прекрасный подарок любителям адвентюр и игр жанра AD&D, подготовленный В. Овечкиным из С. Петербурга. Кроме того, наполовину подготовлена очередная книга ZX FORUM 4, готовится рукопись книги Stalker'a (г. Харьков) с примерным названием "ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ НА ZX SPECTRUM", обещал написать книгу KSA из Москвы... Для успешной и скорой реализации этих планов нужна Ваша заинтересованная поддержка, наш уважаемый читатель!

Как Вы знаете, другим видом нашей деятельности является развитие направления "АВТОРСКАЯ ПРОГРАММА". Следует сказать, что, несмотря на большие трудности в преодолении "совкового" отношения к авторским правам, нам удалось заложить основы цивилизованного рынка программ отечественных авторов. В связи с этим мы намерены всемерно развивать и поддерживать отечественных авторов как игровых, так и системных программ; готовы принимать Ваши работы на реализацию с выплатой авторского вознаграждения. Просим ВСЕХ авторов качественного программного обеспечения присылать нам как свои новые работы, так и программы, уже имеющие распространение - мы обеспечим Вам всероссийскую известность.

Теперь обращаемся ко всем фирмам, хакерским группам и другим лицам, занимающимся распространением авторских программ, исключительные права на которые имеет "ИНФОРКОМ". До настоящего времени мы сквозь пальцы смотрели на этот процесс. Однако к нам поступает все больше писем, в которых читатели и авторы программ указывают на недопустимость этого явления.

Поэтому, ставя на первое место интересы авторов и пользователей, мы предлагаем всем заинтересованным в распространении программ фирмам и лицам, войти с нами в контакт для развития ВЗАИМОВЫГОДНОГО сотрудничества.

Практически все, кто продает наши программы, нам известны. Те из них, кто будет и далее беззастенчиво продолжать торговлей чужим товаром, станут «известны» всей стране - мы будем периодически публиковать их списки (просим наших читателей помочь с их составлением). Кроме того, тот, кто не соблюдает чужих авторских прав, не вправе рассчитывать на то, что будут соблюдаться его авторские права. Это относится, в первую очередь, к Москве и Санкт-Петербургу, а также, к другим городам и весям... Напоминаем, что нелегальное копирование на продажу чужих авторских программ является САМОЙ БОЛЬШОЙ опасностью дня зарождающегося рынка программного обеспечения на ZX SPECTRUM.

Итак, ждем Ваших предложений по адресу 121019, Москва, а/я 16 или по телефону из текущего Прайс-листа (в указанный там день), или лично по адресу нашего корпункта (Новый Арбат, д.2).

ПРИГЛАШАЕМ К СОТРУДНИЧЕСТВУ:

Разработчиков оригинальных игровых и системных программ. Берем на себя функции по продвижению их на рынок. Авторам выплачивается вознаграждение

АВТОРСКАЯ РАЗРАБОТКА

© С. Веремеенко

АДАПТАЦИЯ ИГР 8-БИТОВЫХ ВИДЕОПРИСТАВОК ДЛЯ ZX-SPECTRUM С ВИДЕОПРОЦЕССОРОМ

В ZX-Ревю 1995/6 была опубликована статья «**Видеопроцессор**» для ZX-Spectrum". Первые отклики на нее показали, что наибольший интерес (пока!!) вызвала возможность адаптации игр 8-битовых видеоприставок для ZX-Spectrum с видеопроцессором.

Это несколько не тот эффект, на который я рассчитывал, но, как говорится: "Голос народа - глас божий".

Вообще-то, при разработке интерфейсной платы не ставилась задача возможности адаптации дендивских игрушек под Spectrum. Это выявилось уже при завершении работы, как некое бесплатное приложение.

Чтобы расширить круг программ, адаптация которых возможна, пришлось внести небольшие изменения в схему интерфейсной платы. Новая схема полностью совместима со старой при работе в режиме видеопроцессора.

Её дополнительные функции используются только при загрузке дендивских игрушек.

Схема нового варианта интерфейсной платы приведена на рис. 1. Микросхемы D10 и D14 можно заменить на 556PT11. В этом случае резисторы R1-R8 устанавливать не нужно. Если DENDY подключается по низкой частоте, то ее блок питания можно не подключать и, в любом случае, выключатель питания должен быть в положении "выключено", т.к. питание заведено от SPECTRUMa.

Прошивки ПЗУ приведены в таблицах на рис. 2-4.

Ограничения на адаптацию.

Интерфейсная плата в своей максимальной конфигурации, имеет один банк ОЗУ программ емкостью 32 К. Некоторые игры занимают несколько (до 8) банков. Если в игре используется быстрое переключение банков, такая игра, для предложенного варианта интерфейсной платы, адаптирована быть не может. Не представляет особого труда увеличить емкость ОЗУ до 128 К. При этом появится возможность адаптации любых игр, но вряд ли это целесообразно. Стоимость платы резко возрастет, а количество игр увеличится не очень заметно. Подавляющее большинство игр из кассет могут быть адаптированы под существующий вариант.

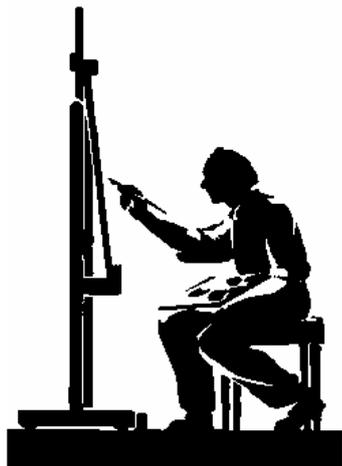
Существуют несколько разновидностей кассет 8-битовых видеоприставок. Рассмотрим некоторые из них.

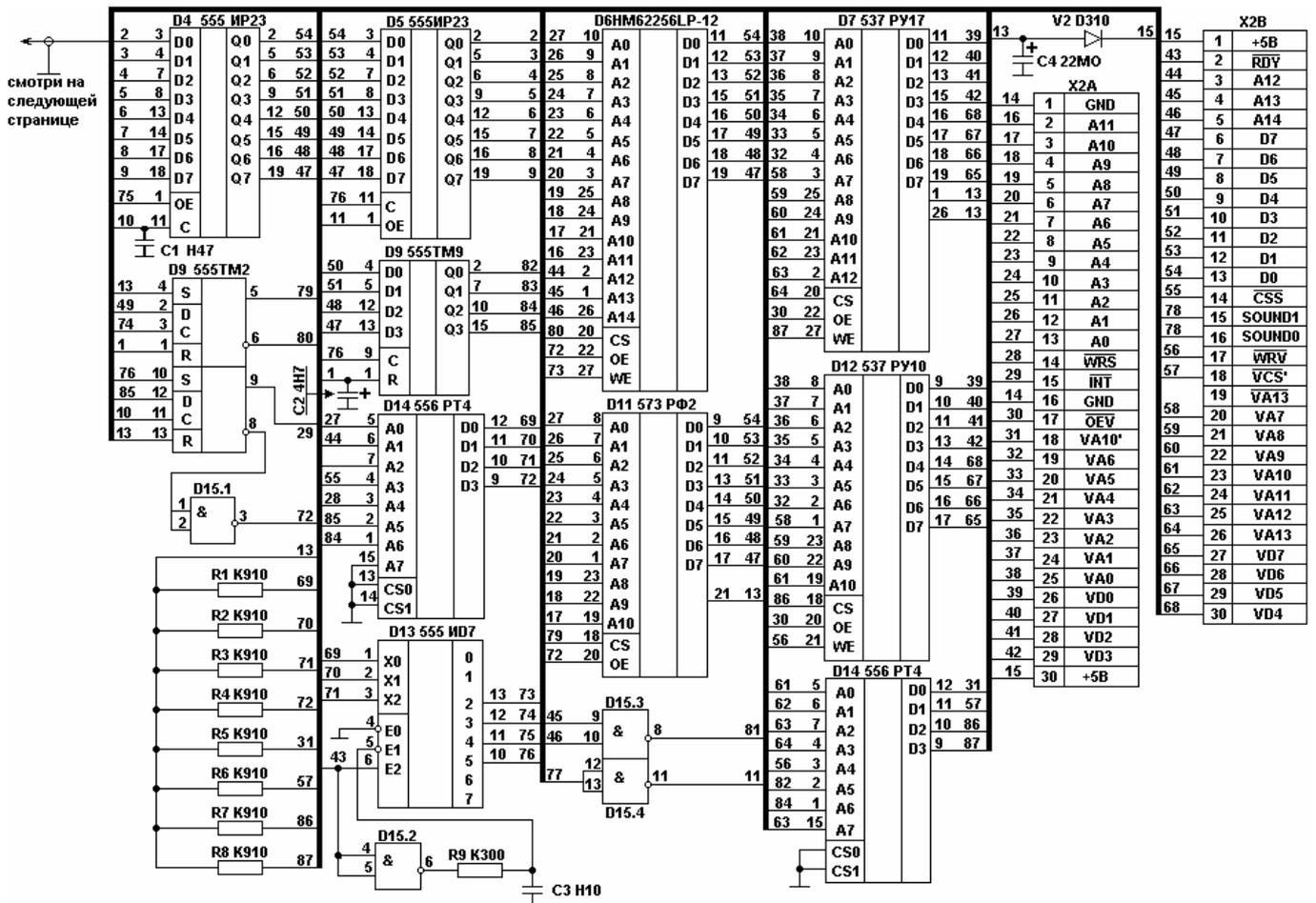
В простейшем и самом приятном случае, кассета содержит 1 банк ПЗУ программ (в дальнейшем S-ROM) и 1 банк видеоПЗУ (V-ROM). Эти кассеты обычно содержат одну игру. Считывание информации из них не представляет никаких трудностей. В принципе, можно использовать любой программатор ПЗУ с соответствующим переходником. Файлы S-ROM и V-ROM в коррекции не нуждаются, и их достаточно снабдить BASIC-загрузчиком.

Наиболее распространены кассеты, в которых ПЗУ делится на несколько банков. Банки ПЗУ переключаются путем записи кода в специальный регистр, как правило, выполненный на микросхеме 74ALS161 (1533IE10). Эта микросхема используется как регистр с параллельной записью. При считывании информации из таких кассет возникают некоторые трудности.

Запись в регистр осуществляется сигналом WRS, формально- это запись в ПЗУ. Естественно, никакой записи в ПЗУ при этом не происходит, но кассеты устроены так, что S-ROM по сигналу записи переходит в режим чтения! Если порт, выставивший данные, достаточно мощный, он, конечно, передаст достаточно хилый S-ROM, но, в этом случае, невозможно поручиться за сохранность кассеты. Ну, а если порт достаточно деликатный, байт на входе регистра банков выставит S-ROM, и что туда запишется предсказать невозможно. Единственный выход заключается в том, что в S-ROM нужно записывать то, что уже там есть. Положение усугубляется тем, что входы регистра банков могут быть подключены как к шине данных, так и к шине адреса, и разряды могут использоваться любые. Кроме того, часто используется дополнительная логика, которая может модифицировать номер банка в зависимости от других разрядов. А если учесть еще и то, что иногда применяются бескорпусные микросхемы, залитые эпоксидкой, и проследить схему по дорожкам невозможно.

Однако, положение не так безнадежно, как может показаться. Мною написана программа "Анализатор кассет V1.4" Она позволяет за несколько секунд определить все параметры кассеты этого типа, считать информацию и записать ее в виде группы файлов на дискету. Кассета подключается к компьютеру через специальный адаптер схема которого приведена на рис 5.

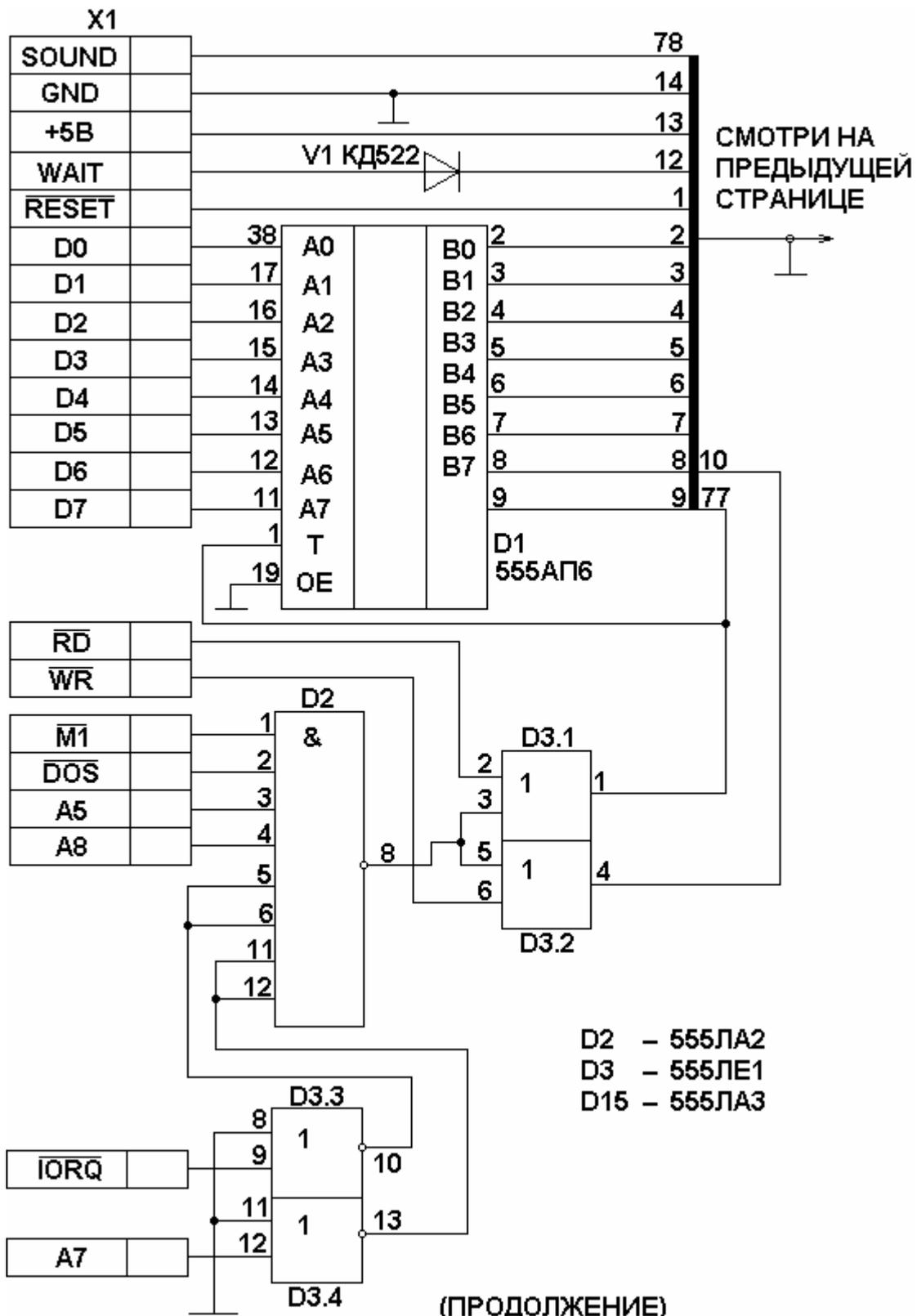




Адаптер содержит 4 выходных и 2 входных порта, буфер данных и дешифратор адресов портов. Если у Вас возникнет необходимость изменить адреса портов, то придется изменить схему дешифратора. Через порты ADRF и ADRL устанавливается адрес S-ROM или V-ROM, через порты, RDAT и WDAT производится чтение и запись данных. Порт WDAT имеет третье состояние выхода. Порт UPP служит для передачи управляющих сигналов. Порт РЕЖ позволяет определить режим работы V-ROM данного картриджа.

Некоторую сложность представляет выбор адресов портов. В разных моделях SPECTRUM для дешифрации системных портов задействованы различные разряды адресной шины с различной степенью неполноты дешифрации. Вследствие этого очень трудно выбрать адреса внешних портов, которые не конфликтовали бы с внутренними во всех моделях. Чтобы не заставлять пользователей переделывать свои компьютеры, программа «Анализатор картриджей» позволяет выбрать произвольные, наиболее удобные для каждой модели адреса портов. Исходно, в ней используются адреса 04h, 0Ch, 14h, 1Ch. Эти адреса внешних портов удобны для использования в SCORPION ZS-256. При необходимости их изменения нужно выбрать в главном меню пункт "НАСТРОЙКА ПОРТОВ" и внести в таблицу портов соответствующие изменения. Программа с внесенными изменениями перезапишется на дискету, и в дальнейшем будет использовать, указанные Вами адреса портов. Эту процедуру можно проводить многократно, но если у Вас не очень надежный дисковод, позаботьтесь предварительно о страховой копии программы. Разумеется, «Анализатор картриджей» не всемогущ, и если получена диагностика «Анализ невозможен» или явно бессмысленный результат анализа, то остается только с вздохом отложить картридж. Я проанализировал несколько десятков картриджей. «Не по зубам» анализатору пришлось 5 или 6.

Есть еще один тип картриджей, в которых функцию переключения банков выполняет специальная заказная микросхема. Структуру таких картриджей можно определить обычно только методом последовательных приближений, считывая очередной банк и определяя условия переключения на следующий путем анализа программы, записанной в нем. К счастью, таких картриджей немного, и обычно проще найти другой картридж с более простой структурой, содержащий те же игры.



Следующий этап адаптации, после того как структура картриджа определена и информация из него перекачана на дискету, чем-то напоминает процедуру дискофицирования ленточных программ для Spectrum. Возможны два варианта. Можно разделить все игры из картриджа и адаптировать каждую отдельно, а можно адаптировать картридж целиком. Есть достоинства и недостатки у обоих вариантов. Раздельная адаптация, в большинстве случаев, проще и позволяет сделать ленточную версию. Адаптация всего картриджа позволяет сохранить меню картриджа и все варианты игр, что при раздельной адаптации обычно теряется. Рассмотрим вначале раздельную адаптацию.

В большинстве случаев каждая игра занимает один банк S-ROM и один V-ROM. Соответствие банков определяется достаточно просто, если «Анализатор» показал одни и те же разряды регистра банков для S-ROM и V-ROM. В этом случае файлы на дискете, относящиеся - к одной игре, будут иметь один и тот же номер. Если разряды регистра банков различны для S-ROM и V-ROM, то соответствие можно определить, просматривая файлы S-ROM кроссмонитором MON6502, который имеет команду поиска команд переключения банков. Когда все команды переключения найдены, определяем соответствие битов кодам банков.

Например, "Анализатор" выдал следующий диагноз:

СЕЛЕКЦИЯ БАНКОВ ПО ДАННЫМ БАНКОВ S-ROM ЕМКОСТЬЮ 16KB

04

БАНКОВ V-ROM ЕМКОСТЬЮ 8KB

04

КОДЫ БАНКОВ:

<i>S-ROM00</i>	<i>00</i>	<i>V-ROM00</i>	<i>00</i>
<i>S-ROM01</i>	<i>04</i>	<i>V-ROM01</i>	<i>01</i>
<i>S-ROM02</i>	<i>08</i>	<i>V-ROM02</i>	<i>02</i>
<i>S-ROM03</i>	<i>0C</i>	<i>V-ROM03</i>	<i>03</i>

РЕЖИМ A11

Из этого следует, что банки S-ROM переключаются разрядами D2 и D3, а банки V-ROM разрядами D0 и D1.

Допустим, MON6502 нашел команду переключения банков:

FF52 LDA #FAh

FF54 STA #FF53h

В аккумулятор загружается число FAh и производится запись по адресу FF53, который соответствует второму байту команды загрузки аккумулятора. Это почти стандартный прием. 4 старших бита значения не имеют, а 4 младших вызовут включение S-ROM02 и V-ROM02.

Хорошие результаты может дать метод «научного тыка», когда на заключительном этапе адаптации, просто случайным образом объединяются разные S-ROM и V-ROM. При работе с дисководом это занимает меньше времени, чем анализ программы.

Следующий этап заключается в восстановлении программы. Это самый сложный этап, и результат здесь во многом зависит от Ваших хакерских способностей. Если в картридже всего одна игра или игра не имеет модификаций, то этот этап пропускается. Если же модификации есть, например, в меню фигурируют MARIO, MARIO1, ... MARIO999, игру придется восстанавливать. Все эти модификации делаются довольно просто. В любой игре есть процедура инициализации, которая устанавливает значения переменных. В лучшем случае, в программе, обслуживающей меню, есть своя процедура инициализации, а основная программа запускается не с начала, хотя присутствует в S-ROM полностью и в неискаженном виде. А в худшем... Китайские хакеры вообще работают очень грубо. Что они делают с играми «при помощи лома и кувалды» словами не передать. Впрочем, с грубыми и однообразными приемами, при некотором навыке, справляться легче.

Не забудьте установить вектор "START" на начало игры. Пока существует единственный инструмент для просмотра и внесения изменений в программы для DENDY. Это кроссмонитор MON6502.

Он имеет следующие команды:

L - загрузка файла S-ROM. На экране появляется имя первого файла на дискете.

CS+6 и **CS+7** - выбор файла,

ENTER - загрузка.

S - сохранение измененного файла. Новый файл не создается, изменения вносятся в исходный.

Z - дизассемблер. Подрежимы:

R - новый адрес,

P - РОКЕ адрес, данные;

ПРОБЕЛ – следующая страница,

CS+0 - выход в меню

D - дамп. Подрежим:

P - РОКЕ адрес, данные,

ПРОБЕЛ – следующая страница,

CS+0 - выход в меню.

T - текстовый дамп.

ПРОБЕЛ – следующая страница,

CS+0 - выход в меню.

F - поиск

ПРОБЕЛ – следующий элемент поиска,

CS+0 - выход в меню.

Поиск имеет 4 модификации:

Тип 1 - поиск команд переключения банков. Ищутся команды записи в область 8000h-OFFFFh.

Поиск проводится по формальным признакам, поэтому монитор может дать «ложную тревогу». Окончательно решать: найдена ли нужная команда или это случайное сочетание байтов - Вам.

Тип 2 - поиск обращений к видеопроцессору. Запись и чтение по адресам 2000h-2007h.

Тип 3 - поиск обращений к портам. Запись и чтение по адресам 4000h-4020h.

Тип 4 - поиск последовательности от 1 до 3 заданных байтов.

Искомые байты вводятся через запятую. Во всех режимах найденная последовательность выделяется повышенной яркостью. Все аргументы команд вводятся в шестнадцатеричной системе счисления.

Некоторые рекомендации.

Хотя маскируемое прерывание в играх не используется, его вектор часто указывает на истинное начало игры, размещенной в данном банке. Вектор START показывает на переход к меню. Очень часто, но не всегда, стартовый адрес совпадает с началом банка. Достаточно надежно начало игры можно найти, задав поиск команды установки стека, если ее не «ампутировал» китайский хакер.

Если в начале игры обнаружилось длинные цепочки NOP, значит игра запускается с начала, но инициализация неполная. В этом случае недостающие команды несложно найти в программе МЕНЮ.

Гораздо хуже, если начало игры полностью разрушено. В этом случае Вам понадобится поистине ангельское терпение и весь отпущенный природой интеллект. При отдельной адаптации игра полностью загружается в ОЗУ интерфейсной платы, и в дальнейшем никакого взаимодействия между Spectrum и DENDY нет вплоть до сброса.

При адаптации картриджа целиком, необходимо найти в памяти DENDY свободный участок около 200 байт. В нем нужно разместить небольшую подпрограмму, которая будет выполнять функцию диспетчера. Алгоритм загрузки приблизительно таков:

1. Загружаем банк памяти, содержащий МЕНЮ и устанавливаем вектор маскируемого прерывания на диспетчер.

2. Порт 6000h устанавливаем в 00h и запускаем МЕНЮ. SPECTRUM непрерывно читает порт 7Fh, и пока в нем 00h, не предпринимает никаких действий.

3. Выбираем пункт МЕНЮ с нужной игрой, при этом в порт 6000h записывается код игры.

4. SPECTRUM, прочитав код игры, считывает соответствующий блок с дискеты и загружает его в DENDY.

Аналогичный алгоритм можно использовать при адаптации игр, занимающих несколько банков, когда используемые банки меняются при переходе на следующий уровень или в другую локацию. К сожалению, он малопригоден для компьютера без дисковода.

Организация взаимодействия между SPECTRUM и DENDY может потребоваться и в еще одном случае. Не очень удобно, кроме клавиатуры и джойстика SPECTRUMа использовать еще и DENDY'ский джойстик. Можно заменить подпрограмму чтения состояния DENDY'вского джойстика подпрограммой, принимающей управляющий код от SPECTRUMа по прерываниям. В этом случае DENDY'вской игрушкой станет возможным управлять с клавиатуры или KEMPSTON-джойстиком. Адреса портов джойстиков в DENDY 4016h и 4017 h MON6502 позволяет найти подпрограмму чтения джойстика за несколько секунд. Загрузка игры.

В ПЗУ интерфейсной платы размещена программа-загрузчик. Предусмотрено 4 режима загрузки. Три из них упрощенные, рассчитанные на применение стандартных загрузчиков. Комплект загрузчиков для всех возможных вариантов предложен мною на дистрибуцию в «Инфорком». При использовании стандартного, загрузчика заключительный этап адаптации упрощен до предела. Предположим, нужно перевести на дискету игру с S-ROM емкостью 32 Кб и режимом видеопроцессора A10. Объединяем файлы V-ROM и S-ROM в один (V-ROM при объединении должен занимать в памяти компьютера младшие адреса) размером 40 Кб и снабжаем его BASIC-загрузчиком D32A10. Все стандартные загрузчики универсальны, они одинаково позволяют загрузить игру с дискеты и кассеты.

Четвертый режим существенно сложнее, но он позволяет использовать все возможности интерфейсной платы. В основном он предназначен для загрузки программ, в которых во время работы сохраняется возможность взаимодействия SPECTRUM и DENDY.

В четвертом режиме в память DENDY по адресу 0700h загружается загрузчик в машинном коде DENDY длиной 256 байт, и управление передается ему. За универсальность приходится платить необходимостью писать программу на непривычном для синклериста ассемблере 6502. Для перевода загрузчика в 4 режим, перед передачей 256-байтного блока нужно передать байт 01h.

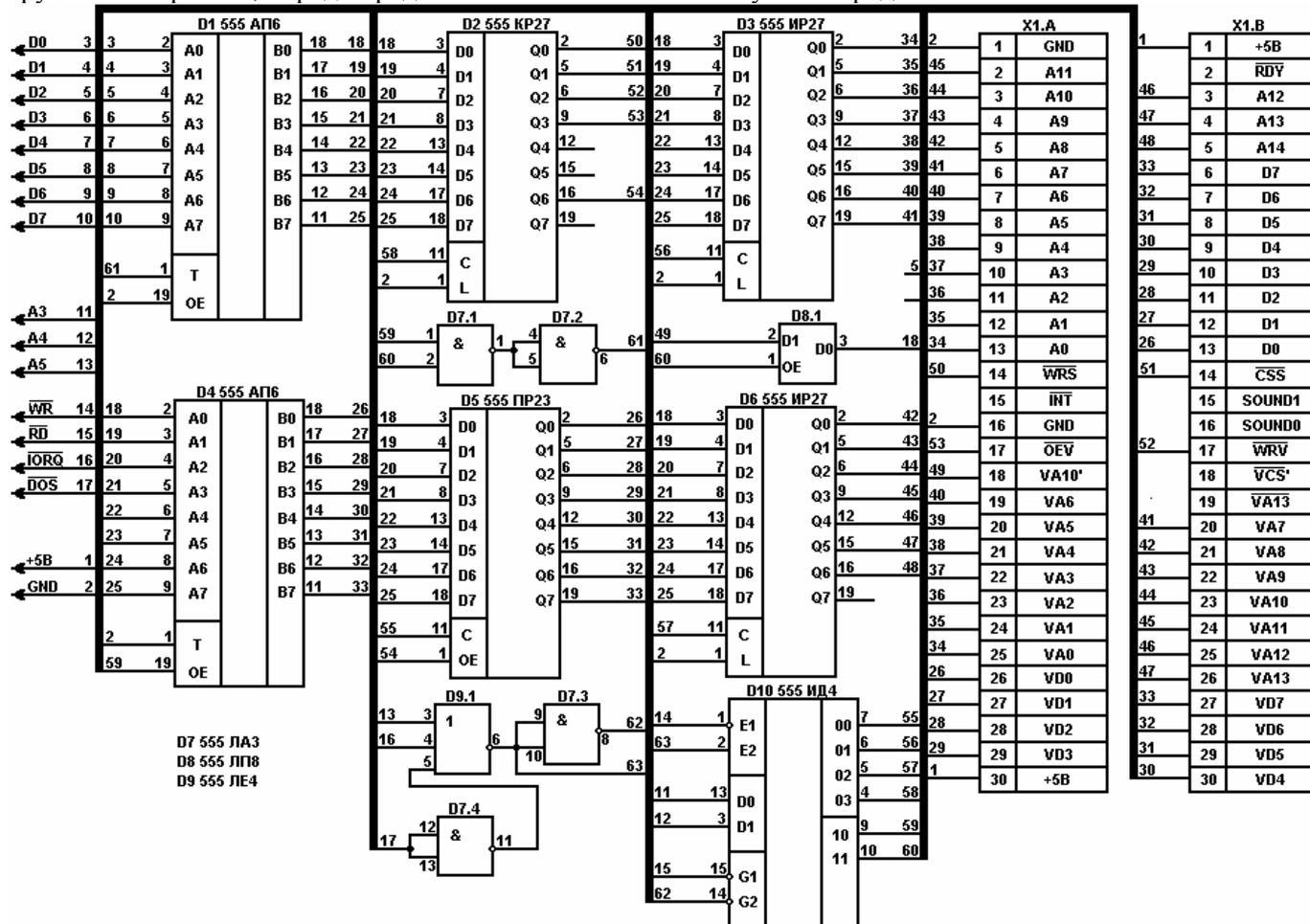


Рис. 5

Прошивка микросхемы D10

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	0A	0D	0B	0F	0F	08	08	08	08							
01	00	00	00	00	00	00	00	00	0C	08	0E	0E	08	08	08	08
02	0A	0D	0F	0F	0F	08	08	08	08							
03	00	00	00	00	00	00	00	00	0C	08	0E	0E	08	08	08	08
04	08	08	08	08	08	08	08	08	0D	0B	08	08	08	08	08	08
05	00	00	00	00	00	00	00	00	0C	08	0E	0E	08	08	08	08
06	08	08	08	08	08	08	08	08	0D	0F	08	08	08	08	08	08
07	00	00	00	00	00	00	00	00	0C	08	0E	0E	08	08	08	08

Рис. 2

Прошивка микросхемы D14

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	06	06	06	06	06	06	06	06	0C	0D	0A	0B	0E	0E	0E	0E
01	0E	0C	0D	0A	0B	0E	0E	0E	0E							
02	06	06	06	06	06	06	06	06	0C	0D	0A	0B	0E	0E	0E	0E
03	0E	0C	0D	0A	0B	0E	0E	0E	0E							
04	0E	0C	0D	0A	0B	0E	0E	0E	0E							
05	0E	0C	0D	0A	0B	0E	0E	0E	0E							
06	0E	0C	0D	0A	0B	0E	0E	0E	0E							
07	0E	0C	0D	0A	0B	0E	0E	0E	0E							
08	06	06	06	06	06	06	06	06	0C	0D	0C	0D	0C	0D	0C	0D
09	0E	0C	0D	0C	0D	0C	0D	0C	0D							
0A	06	06	06	06	06	06	06	06	0C	0C	0D	0D	0C	0C	0D	0D
0B	0E	0C	0C	0D	0D	0C	0C	0D	0D							

0C	0E	0C	0D	0C	0D	0C	0D	0C	0D							
0D	0E	0C	0D	0C	0D	0C	0D	0C	0D							
0E	0C	0C	0D	0D	0C	0C	0D	0D								
0F	0E	0C	0C	0D	0D	0C	0C	0D	0D							

Рис. 3.

Прошивка микросхемы D11

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
F800	78	D8	A9	00	8D	00	60	8D	00	20	8D	01	20	AD	02	20
F810	10	FB	AD	02	20	30	FB	A9	00	8D	00	20	8D	01	20	A9
F820	3F	8D	06	20	A9	F0	8D	06	20	A9	0D	8D	07	20	A9	FF
F830	8D	17	40	A2	96	AD	02	20	10	FB	AD	02	20	30	FB	CA
F840	D0	F3	A0	00	98	99	00	00	C8	D0	FA	A0	00	B9	00	F9
F850	99	00	03	C8	D0	F7	B9	00	F9	D9	00	03	D0	ED	C8	D0
F860	F5	4C	00	03	FF											
Далее до адреса F8FF заполнено FF																
F900	A9	20	8D	01	60	A9	3F	8D	06	20	A9	F0	8D	06	20	A9
F910	10	8D	07	20	A2	FF	9A	A0	00	8C	06	20	8C	06	20	A9
F920	80	8D	00	80	A9	20	8D	00	E0	A9	08	8D	00	F8	AD	00
F930	80	8D	00	60	A9	03	8D	FB	FF	8D	FD	FF	8D	FF	FF	A9
F940	51	8D	FE	FF	A9	62	8D	FB	FF	58	EA	EA	EA	78	4C	49
F950	03	AD	00	60	C9	01	F0	0B	85	20	A9	68	8D	FE	FF	8D
F960	00	60	40	A9	7A	4C	6F	F9	AD	00	60	85	23	A9	8D	8D
F970	FE	FF	A2	00	A0	00	8D	00	60	40	AD	00	60	99	00	07
F980	C8	F0	04	8D	00	60	40	CA	9A	58	4C	00	07	AD	00	60
F990	8D	07	20	E8	D0	0C	C8	C0	20	D0	07	A0	00	A9	A6	8D
F9A0	FE	FF	8D	00	60	40	A5	23	C9	FF	D0	10	C0	FE	D0	0C
F9B0	AC	00	60	A9	C8	8D	FE	FF	8D	00	60	40	AD	00	60	91
F9C0	22	C8	D0	DE	E6	23	D0	DA	AD	00	60	8D	FF	FF	8D	00
F9D0	60	8C	FE	FF	A5	20	F0	11	A2	E0	C9	FF	F0	08	A2	F0
F9E0	C9	FE	F0	02	A2	F8	8E	01	60	6C	FC	FF	FF	FF	FF	FF
Далее до адреса FFF0 заполнено FF																
FFF0	FF	62	F9	00	F8	62	F9									

Рис. 4.

ВИНЧЕСТЕР ДЛЯ ZX SPECTRUM.

Давно назрел вопрос о подключении винчестера к ZX SPECTRUM. Известно несколько попыток решения этой проблемы (МФК: в частности, см. ZX РЕВЮ 96/3). Видимо, разработки в этой области будут продолжаться, и это грозит еще большей неразберихой, чем несовместимость разных моделей SPECTRUM'a, если не будет принят какой-то стандарт.

Цель этой статьи - предложить стандарт подключения винчестера к ZX SPECTRUM и призвать разработчиков следовать ему.

Для начала сформулируем несколько постулатов.

1. Контроллер винчестера должен быть пригоден для любой модели ZX SPECTRUM без каких-либо переделок как компьютера, так и контроллера.
2. Контроллер винчестера должен подключаться к разъему дисководов компьютера (системный разъем есть не у всех моделей). Использование дополнительных сигналов и, упаси боже, портов недопустимо.
3. Любая дисковая версия программы должна работать с винчестером без всякой адаптации.
4. Никаких ограничений на использование альтернативных DOS (IS DOS, CP/M) быть не должно.
5. Размещение драйвера винчестера в ОЗУ компьютера и изменение его системных переменных недопустимо.
6. Способ передачи служебной информации в контроллер винчестера и команды должны быть стандартизованы.

Требования достаточно жесткие, но их выполнение гарантирует отсутствие «головных болей» у синклеристов, подключивших винчестер к своему компьютеру.

Единственный разумный способ удовлетворить все эти требования - работа контроллера винчестера в режиме эмуляции дисковода. При этом, разумеется, теряется такое преимущество винчестера, как более высокое, по сравнению с дисководом, быстродействие. Однако, преимущества такого решения - простота, подключения, независимость от типа компьютера, абсолютная программная совместимость - с лихвой перекрывают этот недостаток. Кроме того, SPECTRUM – это не IBM, и файлов с объемом более 40-50Кб у него не бывает. Поэтому быстродействия дисковода, как правило, вполне достаточно.

В подавляющем большинстве моделей ZX SPECTRUM основной является дисковая система TR DOS, разработчики которой явно не предполагали, что их детищу придется обслуживать винчестер. На жестком диске может храниться несколько десятков тысяч файлов и, без иерархической структуры каталогов, работать с ними невозможно. Эта проблема имеет достаточно простое решение.

Винчестер разбивается на разделы, каждый из которых имеет объем 640Кб и воспринимается TR DOS как дискета. В нулевом разделе, который включается при первом обращении к TR DOS, размещаем файловую оболочку, внешне напоминающую знаменитый NORTON-COMANDER. Она имеет собственную систему каталогов и, перед выбором заказанного пользователем файла, выбирает соответствующий раздел. Есть, правда, одно ограничение. Если программа в процессе работы обращается к дисководу, все дополнительные файлы должны размещаться в том же разделе.

Встает вопрос, а как переключать разделы? Мы ведь договорились, что никаких дополнительных портов и линий связи, кроме тех, которые используются для подключения дисководов, не используем. Собственно говоря, выбор весьма невелик

Из всех линий интерфейса дисковода доступны для управления только четыре линии "выбор дисковода" и "сторона". Дисководы А и В подключены во многих компьютерах, и эти линии использовать нельзя. Вместо дисковода С подключается контроллер винчестера. Я предлагаю для передачи команд контроллеру винчестера использовать линии «дисковод D» и «сторона». Выбор дисковода D означает, что контроллер винчестера должен принять команду, а информация передается по шине «сторона» в последовательном коде. На первый взгляд, это напоминает «операцию через совсем другую полость», но на практике оказалось достаточно удобно.

Предлагается следующий протокол:

1. Выбирается «дисковод D», «сторона» = 1 на время не менее 10 mS.
2. По шине «сторона» передается калибровочный нулевой импульс длительностью $T=5$ mS. Калибровка необходима, т.к. быстродействие у разных компьютеров различно (имеется ввиду режим Турбо).
3. После паузы, длительностью $3*T$, передается код операции в последовательном асинхронном 8-битном коде с битом контроля по четности и двумя стоп-битами. Каждый бит имеет длительность T.
4. Восстанавливается состояние системного регистра дискового интерфейса.

Команды контроллера винчестера:

1. **20h** - выбор раздела. Второй байт - номер раздела.
2. **21h** – принудительная парковка винчестера.

3. **22h** - по этой команде винчестер получает имя А, а дисковод А имя С. Эта команда введена из-за того, что некоторые программы не хотят работать с дисковода, отличного от А.

4. **23h** – восстановление родных имен.

5. **24h** - побитовое копирование с дисковода А на винчестер.

6. **25h** - побитовое копирование с винчестера на дисковод А.

Последние две команды позволяют перенести на винчестер целиком дискету с практически любой защитой и сделать для себя страховую копию такой дискеты. О другом применении этих команд умолчу - это уже вопрос Вашей совести.

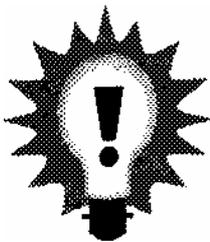
Я понимаю, что предлагать свою разработку в качестве стандарта нескромно и самонадеянно. Возможно, появятся и лучшие, но пока я таких не знаю.

Контроллер предназначен для работы с винчестерами MFM/RLL, имеющими интерфейс ST506/412. Эти винчестеры достаточно широко распространены, и в 2-3 раза дешевле винчестеров IDE такой же емкости. Реально, цена винчестера вместе-с контроллером близка к цене хорошего импортного дисковода.

По мере удешевления винчестеров IDE, возможно, встанет вопрос о разработке контроллера, совместимого с уже разработанным, и для них.

По вопросу приобретения контроллера винчестера или комплекта для самостоятельной сборки, включающего печатную плату, запрограммированное ПЗУ, дискету с инсталлятором, схему и инструкцию, можете обратиться по телефону 3432-340-124.

НОВЫЕ ПРОГРАММЫ



Сегодня мы предлагаем Вашему вниманию две программы из Харькова. Очевидно, Вам не нужно представлять Stalker'a, чей монитор STS уже давно является лучшим программным продуктом в этой области. Наконец-то вышла новая, кардинально переработанная версия этой замечательной программы.

Вторая программа ALASM 128/512 принадлежит перу нового для нас автора – Алексея Мищенко. Это новый ассемблер, отличающийся в лучшую сторону от имеющихся на рынке и, что очень важно, предназначенный для работы в паре с STS 4.2. Итак, слово авторам.

STalker Stealth monitor. Версия 4.2. (STS v.4.2) © STALKER, г. Харьков.

С момента выхода в свет версии STS 3.3 прошло уже довольно много времени, за которое было выпущено несколько промежуточных и одна завершенная версий моего монитора. Впрочем, по порядку.

Версия STS 4.0.

STS 4.0 является демонстрационной версией 4-й серии мониторов.

демонстрационная процедура дезассемблирования на диск в формате TASM2.0 (нормально можно дезассемблировать только 1 блок со своей таблицей меток, перед DEFB метки не ставятся);

- новые возможности при отладке в окне Trace (Stop if Dos, Stop if Find, Stop if User);

- использование любой дополнительной страницы ОЗУ (стандартно - 1) для размещения процедур Disasm To Disk и Setup;

- доступ к ПЗУ TR-DOS (Setup);

- куча глюков.

Отличия STS 4.1. (от версий STS 3.X)

- переключение страниц в Резидентах через полную адресацию порта #7FFD, длина Резидента в связи с этим увеличилась и равна 34=#22 байта;

- эмуляция переключения Bank через порты #7FFD, #FD и нестандартные (Trace);

- полная процедура дезассемблирования на диск (Disasm to Disk [D]) до 18 блоков с общей таблицей меток;

- использование старших 6-го и 7-го битов Bank (порт #7FFD) для переключения дополнительных страниц памяти;

- не тормозящие на адресах >#C000=49152 дисковые процедуры;

- новый режим дезассемблирования для ALASM 2.5, позволяющий просматривать на экране текст программы с метками вместо соответствующих чисел, если перед этим в ALASM'e отассемблировать программу (Label_Disasm, [SS+5]);

- запоминание (и переход к ним) 5-и адресов памяти ([A+1..5],[1..5]);

- переход на адрес, выделенный курсором (Put address, [P]), атак же возврат обратно ([SS+P]);

- просмотр дампа памяти из 8байт в верхней строке экрана в режимах Disasm (View dump, [V]);

- поиск до первого, не равного заданному байту (Find: Inverse);

- после ввода строки курсор перемещается на строку вниз;

- исправлена неправильная отладка команд RETI и RETN (регистр R);

- введение стандартного расположения переменных регистров и Memory Adr внутри STS (для всех STS, начиная с 4.1);

- использование процедур опроса клавиатуры ПЗУ BASIC48 (естественно, не портящих ОЗУ, введено для экономии памяти);

- обновленная процедура печати - компрессированный фонт (5*8внутри STS, введен, также, для экономии памяти монитора;

- много ошибок.

Отличия версии 4.2 (от 4.1)

- вывод первых 80 названий файлов каталога в командах Load/Save File;

- команда установки PC на адрес строки с курсором редактирования ([Z]);

- исправлены многочисленные глюки и ошибки;

- устранены некоторые недостатки, а именно:

- опрос клавиатуры с защитой от дребезга;

- правильная запись на диск номера дополнительной страницы;

- правильная работа с 512Кб;

- убран режим «INSERT» во время ввода строки;

- убран автоперевод курсора на строку вниз после ввода.

- поддержка меток ([SS+5]) самого «крутого» на сегодняшний день ассемблера ALASM. Вот в чем его особенности и преимущества (подробно см. ниже):

- это первый ассемблер, поддерживающий не только 128, но и 256 и 512Кб (расширение через #7FFD);
- возможность размещения в любой странице памяти (как самого ассемблера, так и таблиц меток);
- ассемблирование в любую страницу;
- самая высокая скорость ассемблирования за счет однопроходности;
- работа одновременно с несколькими текстами, расположенными каждый в своей странице;
- наличие конвертора из других ассемблеров.

В отличие от предыдущих версий, наконец-то, используется полная адресация (через OUT (C),...), что позволяет использовать STS 4.2 на ЛЮБЫХ компьютерах со стандартным расположением процедур обработки клавиш в ПЗУ BASIC48 и ПЗУ TR-DOS версии 5.03 и выше.

(ИФК: дискриминация владельцев SCORPION'ов закончилась!)

Версия 4.2 является на июнь 1996 года последней версией STS. Приобретение по нелегальным каналам предыдущих промежуточных версий не позволяет владельцу рассчитывать на нормальную работу программы.

STS 4.2 желательно использовать совместно с ассемблером ALASM 2.x.

Обращаю внимание всех заинтересованных лиц:

ВСЕ ПРАВА НА РАСПРОСТРАНЕНИЕ-STС 4.2. ПРИНАДЛЕЖАТ ФИРМЕ "ИНФОРКОМ"!

Теневой макроассемблер-отладчик ALASM 128/512 v.2.5 © А. Мищенко, г. Харьков.

Ассемблер ALASM отличается от остальных оригинальным распределением памяти и тем, что компилирует текст всего за один проход, в результате чего отпадает необходимость загружать включаемые файлы дважды за период компиляции.

Оригинальность распределения памяти заключается в предоставлении пользователю большей свободы выбора относительно того, в каких страницах 128-й памяти хранить свои тексты, так как сам ассемблер занимает только страницу #04, а метки компилируемой программы страницу #03. При этом вся 48-я память остается свободной (страницы #05, #02, #00), а пользователю остаются остальные 3 страницы (#01, #06, #07), в которых может разместиться 48 килобайт сжатого и токенизированного текста.

При условии наличия в памяти монитора STS объем памяти уменьшается до 32 Кб, а при наличии доработки 512 Кб, память текстов практически бесконечна.

Все дело в том, что каждый текст занимает одну страницу памяти, но при этом может сколько угодно раз, и с любой вложенностью вызывать к компиляции тексты, находящиеся в других страницах. При этом загрузка текста с диска производится только в том случае, если этот текст еще не был загружен в необходимую страницу то есть, всего 1 (один) раз за период работы над ним.

Страницы, расположенные в 512-й памяти, не подвержены тестированию при сбросе и не очищаются. Поэтому при работе с расположенными в них текстами вообще отпадает необходимость работы с диском, так как различные trouble'y не приведут к потере текста.

Размер памяти меток ассемблера зависит от количества меток и от длины каждой из них. Это экономит память меток и позволяет разместить в памяти тем большее количество меток, чем меньше их длина. Некоторую экономию памяти меток приносит описание метки до того, как она будет использована в арифметическом выражении.

Большое удобство для пользователя обеспечивается тем, что интерфейс ALASM'a, в основном, совместим с широко известными TASM и MASM, хотя имеет некоторые отличия.

Основным отличием является то, что редактируемое поле экрана содержит матрицу 42*27 символов. Это позволяет сэкономить нервы при визуальной отладке ассемблерной программы, так как на экране отображается одновременно 26 строк (27-я информационная). Длина строки в 42 символа позволяет избавиться от ощущения, что правая половина экрана пропадает зря.

Возможность ввода с клавиатуры букв русского алфавита и символов псевдографики способствует увеличению числа русскоязычных программ красивого оформления, так как лень обычно не позволяет программистам воспользоваться текстовым редактором или списком DEFБ, и поэтому они учат английский язык.

Дополнительные удобства приносят также калькулятор, использующий в своих выражениях метки и арифметические операции над ними, команда поиска места описания метки в тексте, возможность интерактивной отладки текста с помощью монитора STS с выводом меток по соответствующим им адресам и другие полезные мелочи.

Конвертор UNICON позволит перевести исходный текст ассемблера TASM или MASM в формат ALASM, автоматически внося необходимые изменения (однако, кавычки в INCLUDE и INCBIN необходимо расставлять самостоятельно).

TR-DOS для начинающих

Продолжение (начало см. ZX РЕВЮ 96/1-2)

© В. Сироткин, г. Краснокаменск.

ГЛАВА 2.

ВНУТРЕННЯЯ ОРГАНИЗАЦИЯ ДИСКА.

Как вы уже знаете, существует множество систем, форматов и способов записи на диски информации в разных компьютерах.

В системе TR-DOS Спектрума основной стандарт разметки дисков таков:

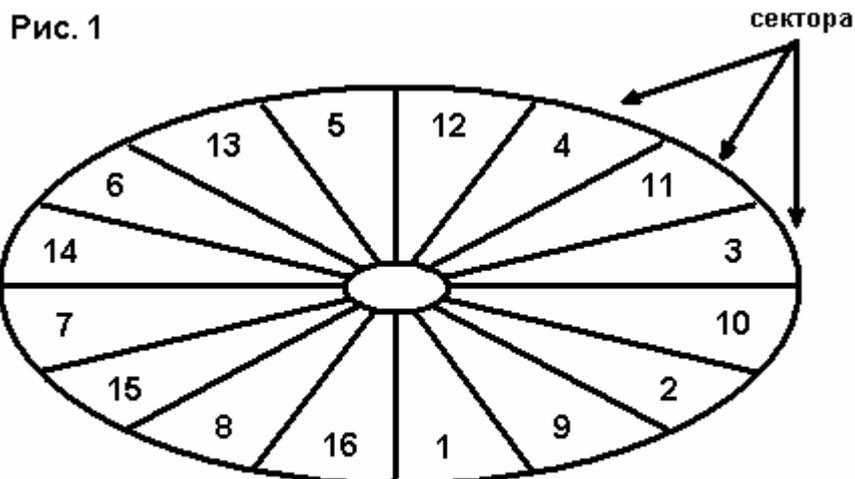
256 БАЙТОВ НА 1 СЕКТОР; 16 СЕКТОРОВ НА ДОРОЖКЕ.

Как уже было сказано, в TR-DOS используются диски DD/DS (2D/2S). Так вот, эти диски TR-DOS форматирует таким образом: 160 ДОРОЖЕК НА ВСЕМ ДИСКЕ, по 80 ДОРОЖЕК НА КАЖДОЙ СТОРОНЕ. Дорожки нумеруются от 0 до 159.

Нулевая дорожка расположена у внешнего края диска на верхней стороне поверхности 0. Далее - 1-я дорожка на нижней стороне диска.

Две дорожки, расположенные на разных поверхностях, но находящиеся одна над другой, называют ЦИЛИНДРОМ. На каждой дорожке расположено по 16 СЕКТОРОВ, которые нумеруются с 1 по 16.

Физически сектора на диске располагаются не по порядку возрастания номеров, а с пропуском в один сектор (см. рис.1).



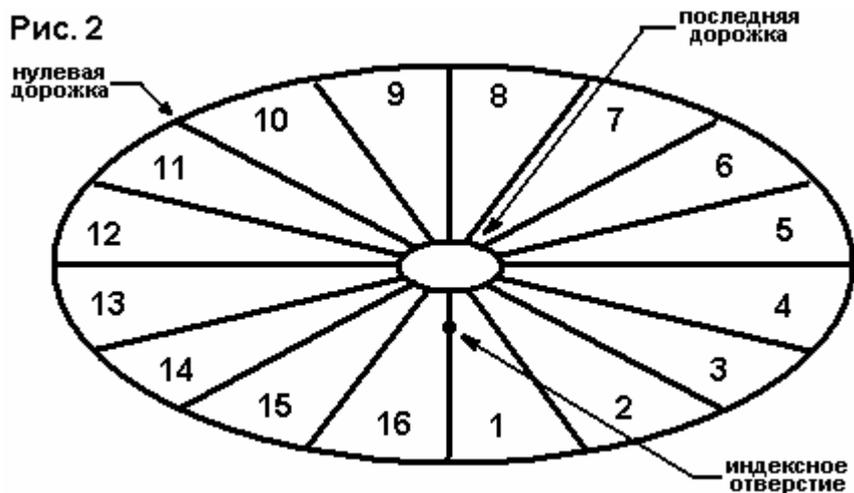
Диск форматирован в TR-DOS командой "FORMAT"

Это было сделано, видимо, из опасения, что процессор, обработав один сектор, не успеет приготовиться и сразу считать следующий сектор. Пришлось бы ждать еще один оборот диска для того, чтобы попасть на следующий по порядку сектор.

А так, пока обрабатывается сектор номер 1, диск прокрутится, пройдет сектор 9, а когда будет все готово, головка дисководов как раз попадет на сектор 2.

Такая разметка диска присуща самой системе TR-DOS, но это не значит, что нельзя разметить по другому. Дело в том, что системе безразлично, вообще-то, как на диске располагаются сектора в дорожке. Самое главное, чтобы на данной дорожке были сектора с нужными номерами, нужной размерности и нужным числом.

Вот г-н Родионов подумал и написал программу DCU 2, где сектора расположены по порядку возрастания номеров, не побоялся, что система не успеет обработать два идущих подряд сектора - и не ошибся! Получился, так называемый, "быстрый FAST диск", где информация с диска обрабатывается чуть быстрее, чем на стандартно размеченном диске (см. рис. 2).



Диск размечен программой Н. Родионова "D.C.U. 2"

Да, и самое главное г-н Родионов не побоялся разметить диск на число дорожек больше, чем 160, и тоже не прогадал, выжав из диска еще дополнительные килобайты.

Вообще-то можно разметить диск на еще большее число дорожек, но чем головка ближе к центру диска, тем больше риск ошибки при считывании информации. И уж совсем плохо будет, если Вам придет в голову разметить диск с двухсотой дорожкой. Ваш дисковод будет безуспешно долбиться головкой об ограничители, и Вам станет не по себе от дикого звука, который при этом будет раздаваться.

Это все хорошо, ну а как, все таки, система узнает, где и в каком месте диска она находится? Дело в том, что у каждого сектора находится служебная область, где записана вся необходимая информация.

В начале у сектора идет область синхронизации, состоящая из кодов #4E и нулей - это, так называемая, область «пробела», нужная для синхронизации микросхемы ВГ93 на область данных (см. Рис.3). Далее идет индексная метка и область И.А.М. (индексная адресная метка). Вот в ней то и записана вся информация о данном секторе. И.А.М. начинается с байтов #A1, #FE. Далее следуют:



Синхро-байты	Метка	Трек	Сторона	Сектор	Длина сектора	Контрольная сумма
Поле синхронизации и идентификации						

Метка адрес	ДАННЫЕ (256 байт)	Контроль
Поле данных с метками и контрольной суммой		

- НОМЕР ЦИЛИНДРА (начало с 0)
- СТОРОНА ДИСКА (0 верх, 1 низ)
- НОМЕР СЕКТОРА (с 1 по 16)
- ТИП СЕКТОРА
 - 0 - сектор 128 БАЙТ (его длина)
 - 1 - сектор 256 БАЙТ
 - 2 - сектор 512 БАЙТ
 - 3 - сектор 1024 БАЙТ

КОНТРОЛЬНАЯ СУММА (циклическая сумма)

Если система не находит область И.А.М., или контрольная сумма не совпадает с просчитанной (а контрольная сумма просчитывается микросхемой каждый раз), то выдается код ОШИБКИ, и обработка или повто-

рывается, или прекращается в зависимости от числа попыток, заложенных программистом в подпрограмму обработки ошибок.

Если все в порядке, то за областью И.А.М. следуют байты пробела, коды #4E и нули. За ними идет АДРЕСНАЯ МЕТКА ДАННЫХ - коды #A1; #FB, и только тогда идут, собственно, те данные, которые считываются или записываются на диск пользователем.

Эта область, когда данным нет, содержит 256 нулей, (в случае TR-DOS). После области данных следует код КОНТРОЛЬНОЙ СУММЫ этой области. И вот так в каждом секторе.

Вся эта информация заносится в процессе форматирования диска и, в конечном итоге, зависит от программиста. Можно разметить по своему усмотрению все сектора на дорожке с одним номером, или дорожки с обратной нумерацией, или еще чего придумать, но тогда придется писать уж и программу, которая будет разбираться на диске со всей этой мешаниной, но это будет уже не TR-DOS, а что-то другое.

TR-DOS в этом не разберется и с пол-оборота выдаст: *DISK ERROR*!

Для примера приводим таблицу всех байтов, которые находятся в одном секторе, в служебной области (см. табл.1).

Таблица 1.

Структура массива одного сектора (дорожка 0, сектор 1)

число байтов	код	назначение
10	#4E	Начальный пробел синхронизации
12	#00	
1	#A1	Начало адресной метки индексного массива
1	#FE	
1	#00	Цилиндр
1	#00	Сторона диска
1	#01	Номер сектора
1	#01	Длина сектора (256 байт)
2	#CRC	Контрольная сумма
22	#4E	Пробел перед областью данных
12	#00	
1	#A1	Начало метки области данных
1	#FB	
256	#DATA	Данные пользователя
2	#CRC	Контрольная сумма
50	#4E	Остаточный пробел
формат массива для диска двойной плотности (МЧМ)		

Вся эта информация заносится за один проход, по всем секторам на одной дорожке во время форматирования.

Собственно, это и есть процесс форматирования. Из этого всего следует, что если у Вас возникнет желание отформатировать одну дорожку - на здоровье, но отдельно отформатировать или переформатировать один сектор нельзя! Только целиком дорожку!

Если мы внимательно подсчитаем, сколько все же килобайт помещается на одной дорожке со всеми метками, «пробелами», и т.д., то окажется - на дорожке приблизительно 6200 байтов:

- 4 килобайта информации, которой располагает пользователь;
- 2 килобайта всех служебных меток.

Да, за все надо платить. Без этих 2-х килобайт меток обмен с диском был бы крайне затруднителен, если возможен вообще!

Итак, у нас есть размеченный диск:

- 160 ДОРОЖЕК , 16СЕКТОРОВ В ДОРОЖКЕ;
- ВСЕ НЕЧЕТНЫЕ ДОРОЖКИ РАСПОЛАГАЮТСЯ НА НИЖНЕЙ СТОРОНЕ;
- НУЛЕВАЯ И ВСЕ ЧЕТНЫЕ НА - ВЕРХНЕЙ СТОРОНЕ.

А что же дальше? Даже если у Вас есть размеченный диск, но не заполнен сектор номер 8, то TR-DOS все равно ответит Вам отказом, и диск рассмотрит как сбойный.

В ВОСЬМОМ ЛОГИЧЕСКОМ секторе располагается информация TR-DOS. Это тип диска, имя диска, количество дорожек и т.д. Ниже дана раскладка всех необходимых байтов сектора.

**СОДЕРЖАНИЕ ВОСЬМОГО (лог.) СЕКТОРА
С нулевого по 224 байт в секторе записаны нули.**

НОМЕР БАЙТА	НАЗНАЧЕНИЕ
225	номер первого свободного сектора
226	номер первой свободной дорожки
227	диск двусторонний/односторонний = #16 \neq #18 =
228	количество файлов на диске
229	количество свободных секторов МЛ. БАЙТ
230	СТ. БАЙТ
231	количество секторов в дорожке = #10
232	00
233	00
234 по 242	#20
243	00
244	количество удаленных файлов
245 по 252	ИМЯ ДИСКА (8 символов)

Самым главным байтом на секторе является байт с номером 231!

Можете стереть или изменить всю информацию на секторе - диск еще останется работоспособным, но если Вы измените этот байт - диск окажется сбойным, и прочитать его можно будет только каким-нибудь Диск-Доктором, загрузив в него целиком дорожку № 0.

ВОТ ТУТ НЕОБХОДИМО СДЕЛАТЬ ОТСТУПЛЕНИЕ И ДОГОВОРИТЬСЯ ОБ ОДНОЙ ВЕЩИ ВО ИЗБЕЖАНИЕ, ПУТАНИЦЫ!! ДЕЛО В ТОМ, ЧТО НОМЕРА СЕКТОРОВ РЕАЛЬНО НА ДИСКЕ ПОМЕЧЕНЫ НОМЕРАМИ ОТ 1 ДО 16!!! СИСТЕМА TR DOS НА ВЕРХНЕМ УРОВНЕ ИСПОЛЬЗУЕТ НУМЕРАЦИЮ С 0 ДО 15, КОРРЕКТИРУЯ ПОТОМ НОМЕРА СЕКТОРОВ ДО РЕАЛЬНЫХ. БУДЕМ СЧИТАТЬ, ЧТО НОМЕРА С 1 ДО 16 БУДУТ ФИЗИЧЕСКИМИ НОМЕРАМИ СЕКТОРА, А НОМЕРА С 0 ДО 15 - ЛОГИЧЕСКИМИ. В КАЖДОМ КОНКРЕТНОМ СЛУЧАЕ МЫ БУДЕМ УКАЗЫВАТЬ, ГДЕ ЛОГИЧЕСКАЯ НУМЕРАЦИЯ, А ГДЕ ФИЗИЧЕСКАЯ.

Восстановить же диск после таких манипуляций можно..., тем же самым «Доктором», записав на это место код #10.

Сектора с 0 по 7 (логические номера) занимает каталог диска.

Пока на диске нет файлов, там записаны нули, но как только на диске появляются файлы, система автоматически создает на каждый файл нечто вроде описателя файла, т.е. создает каталог.

На каждый описатель - заголовок файла резервируется 16 байт. Таким образом, на диске может быть записано не больше (256 байт сектора) / (16 байт заголовка) * (8 секторов) = 128 ФАЙЛОВ!!! Лишние файлы будут просто не создаваться на диске.

Кстати, и название свое система TR-DOS 128 получила именно в честь этого числа.

Каждый байт заголовка имеет свое значение для системы.

Структура заголовка файла имеет следующий вид:

БАЙТЫ	НАЗНАЧЕНИЕ
0...7	ИМЯ ФАЙЛА (8 символов)
8	ТИП ФАЙЛА (В,С,D,#)
9	НАЧАЛО ФАЙЛА (мл. байт) В ОЗУ
10	НАЧАЛО ФАЙЛА (ст. байт) В ОЗУ
11	ДЛИНА ФАЙЛА (мл. байт)
12	ДЛИНА ФАЙЛА (ст. байт)
13	ЧИСЛО ЗАНИМАЕМЫХ СЕКТОРОВ
14	НОМЕР ПЕРВОГО СЕКТОРА ФАЙЛА
15	НОМЕР ДОРОЖКИ НАЧАЛА ФАЙЛА

Имя файла

Может состоять, вообще-то, из любых символов, стрингов, но рекомендуется давать имена, состоящие из кодов не меньше 30 и не больше 127.

Можно, конечно, составить имя и из кодов, не входящих в данный диапазон, но вероятность того, что Вы нарветесь на такую ситуацию, когда после команды "CAT" Ваш компьютер напишет - INVALID COLOR, DISK ERROR и т.д. и т.п. - **ОЧЕНЬ ВЕЛИКА!**

Так что, не искушайте судьбу, да и работать с простыми именами проще.

Тип файла.

Системой воспринимаются только файлы с типом В.

В - файл БЕЙСИКА

С - файл кодов

D - файл типа ДАННЫЕ

- файл произвольно-последовательного типа.

Но это не значит, что нельзя работать с другими типами файлов.

Например, Вам захотелось, чтобы программа в процессе работы загружала или сохраняла файлы состояния ситуации с типом, отличным от системных. Отчего же нельзя, например, дать при выгрузке тип файла < Z >? Можно, но только нужно помнить, что работать с этим файлом из системы TR-DOS стандартными командами Вы не сможете! Это можно сделать только АССЕМБЛЕРОМ.

Начало файла в ОЗУ (СТАРТ).

Это справедливо только по отношению к файлам типа КОДЫ, файл БЕЙСИКА имеет в этих байтах параметр 'ДЛИНА ПРОГРАММЫ'.

Вообще-то, этот параметр нужен только в том случае, когда Вы даете команду: LOAD "filename" CODE без указания - КУДА загружать коды, по какому адресу. Но если Вы даете полное указание, куда загрузить файл с диска, то этот параметр становится не очень то и нужным.

Длина файла в байтах.

Реальная длина файла, то есть, сколько байтов он занимает в ОЗУ. Так как перекачка файла с диска происходит через специальный системный буфер TR-DOS, то эта величина указывает TR-DOS сколько, конкретно, байтов из сектора будет пересылаться по нужному адресу через буфер в ОЗУ.

Количество занимаемых секторов.

Важная величина, по которой и происходит, в конечном итоге, считывание / запись файла целиком. Даже если реальная длина файла составляет 1 байт, все равно на диске он будет занимать целый сектор в 256 БАЙТ.

ОТСЮДА ВЫВОД: желательно, чтобы реальная длина файла при записи, была кратна 256 - тем самым Вы сохраните место на диске, так как сектора будут заняты полностью.

С файлами БЕЙСИКА чуть сложнее, но о них немного позже...

Номер первого сектора файла.

Тут все ясно - номер сектора на дорожке, с которого начинается кодовая последовательность файла. Может принимать значения от 0 до 15 (логический номер). Номер первого трека файла.

Номер дорожки, на которой находится начальный сектор кодовой последовательности файла. Может иметь номера от 1 до 159.

Файлы записываются на диск в виде кодовой последовательности, сектор за сектором. Если на дорожке не хватает секторов, запись переходит на следующую по порядку номеров, дорожку. И так продолжается, пока вся кодовая последовательность не будет записана на диск.

После записи система обновляет, сектор номер 8 и записывает туда первый свободный трек, сектор, количество свободных оставшихся секторов.

Если Вы внимательно прочитали до этого места, то без труда увидите, что длина файла жестко регламентирована объемом в 255 секторов (#FF), т.е. $255 * 256 = 65280$ БАЙТОВ.

Реально, если не пользоваться "переменными" TR-DOS, то можно соорудить файл хоть на весь диск. Естественно, создав такого монстра, Вам потребуется создать и свою программу для его считывания и записи.

Если происходит стирание файла, то, первым делом, в каталоге вместо первого символа имени файла записывается код 01.

В восьмом секторе увеличивается байт "количество стертых файлов".

Реально, после стирания, объем свободных секторов на диске не увеличивается (если, конечно, Вы не стираете самый последний файл в списке - тогда можно не беспокоиться - место освободится).

Чтобы освободить сектора от "стертого файла", необходимо перезаписать все файлы, расположенные после стертого файла, на его место, что и делает команда MOVE. После этой команды происходит перезапись всех файлов, записанных после стертого файла, обновляется информация в секторе номер 8, и у Вас появляется на диске ровно столько свободных секторов, сколько занимал Ваш файл. И расположены эти освобожденные сектора сразу после всех файлов.

В этом одно из неудобств системы TR-DOS: чтобы освободить сектора стертого файла, записанного в начале диска, Вам приходится перезаписывать всю информацию на диске, "сдвигая ее вниз".

Например, в IBM система MS-DOS отслеживает все сектора, и при стирании файлов сектора освобождаются сразу, так что один файл может быть расположен в различных секторах, на различных дорожках и не обязательно в упорядоченном виде.

При загрузке такого файла (его еще называют сильно сегментированным), затрачивается немного больше времени, но система находит по всему диску именно те сектора, на которых записан загружаемый файл.

Но зато у TR-DOS преимущества: ее файлы всегда составляют единое целое и записаны всегда последовательно по возрастанию секторов и треков.

Восстановить файл после процедуры стирания легко - надо найти в секторах каталога Ваш файл с 01 вместо первой буквы и записать вместо 01 какой-нибудь удобоваримый символ, и все!

Да, еще не забудьте уменьшить байт стертых файлов в секторе номер 8 и увеличить байт "количество файлов" на диске.

Но!!! Бели Вы дали команду MOVE, можете не сомневаться - на месте вашего стертого файла будет какой-нибудь другой. И в секторах каталога Вы, скорее всего, не обнаружите даже воспоминания об имени Вашего файла.

Так что НЕ бойтесь стирать, а БОЙТЕСЬ команды: MOVE.

На рисунке 4 представлен дамп первых байтов области каталога диска. Слева представление в шестнадцатеричных кодах справа - в символах. На рисунке 5 дан дамп программы типа 'КОДЫ' в таком виде, в каком он лежит на диске в секторе.

00h..6C6C69732E726F62	llis.rom
08h..43008B0001010001	C
10h..524F424F54524F4E	ROBOTRON
18h..420F010F01020101	B
20h..626F6F7420202020	boot
28h..4255285528290301	BU(U())
30h..544C572E52425420	TLW.RBT
38h..42AE029202030C03	B
40h..746C332B2B202020	tl3++
48h..4386B67A44450F03	C zDE
50h..746C332B20202020	tl3+
58h..4386B67A44450408	C zDE
60h..77737475706C3020	wstupl0
68h..438667E8090A090C	C g
70h..7773747570202020	wstup
78h..438667E8090A030D	C g

Дорожка: 0 Сектор: 0

Рис. 4 Дамп сектора-каталога диска

Имя файла, тип						codefile		.C	
Дорожка						Track	1A		
Сектор						Sect.	4		
18	18	1F	F8	02	40	02	40	?	IOx
3F	FC	30	0C	0F	F0	02	40	?	IOx
0F	F0	10	08	20	04	3C	3C	?	IOx
3C	3C	3C	3C	20	04	10	08	<<<<	<<<<
FF	FF	00	00	00	00	00	00	?	IOx
FF	FF	F8	1F	F8	1F	E0	07	?	IOx
C0	03	C0	03	E0	07	C0	03	?	IOx
E0	01	E0	01	C0	03	F0	0F	?	IOx
E0	07	C0	03	E0	01	E0	01	?	IOx
E0	01	E0	01	E0	01	C0	03	?	IOx
E0	07	F0	0F	FF	FF	FF	FF	?	IOx
00	00	03	C0	02	40	0F	F0	?	IOx
30	0C	3F	FC	02	40	02	40	?	IOx
1F	F8	18	18	07	E0	02	40	?	IOx
0F	F0	10	08	20	04	53	C6	?	IOx
53	C6	53	C6	20	04	10	08	cF	cF

Рис. 5

Давайте теперь посмотрим, как лежит на диске в секторе блок файла, написанного на БЕЙСИКЕ. Составим простую программу:

```
10 PRINT "ROGRAMM NO AUTO RUN"
```

Запишем ее на диск командой:

```
RANDOMIZE USR 15619: REM: SAVE "PROG1"
```

В каком-нибудь ДИСК-ДОКТОРЕ найдем тело самой программы на диске. Найти очень просто: посмотрим сначала область каталога, находим имя нашего файла, находим в заголовке номер первого сектора файла и номер дорожки. Затем Вам нужно будет только ввести эти данные в ДИСК-ДОКТОР, а он уж сам найдет данный сектор на данной дорожке.

Но вот файл на секторе найден, и что же мы видим? (см. рис. 6).

Track	25	Base:	5D3B	to	5D57	
00	0A	18	00	F5	22 50 52	U"PR
4F	47	52	41	4D	4D 20 4E	OGRAMM N
4F	20	41	55	54	4F 20 52	O AUTO R
55	4E	22	0D	80	AA 00 00	UN" *
35	36	31	39	0E	00 00 03	5619
3D	00	3A	EA	3A	F8 22 50	=:j:x"P
52	4F	47	31	22	0D 80 0D	ROG1"
6B	5E	05	00	00	00 00	k↑
5D3B 00						pop

Рис. 6 Дамп программы бейсика без автостарта.

С первыми четырьмя байтами понятно: это номер строки БЕЙСИКА - 2 байта и длина всей строки - 2 байта. Следом идет текст нашей строки, оканчивающийся кодом #0D. Следом идет байт #80, сигнализатор конца всей программы. (На рисунке эти байты выделены инверсией). А дальше идет какой-то довесок. Мы видим справа в символьном представлении и имя нашей программы и, вроде бы, число 15619, и многое другое.

Все правильно! Вся эта информация была необходима TR-DOS на этапе работы, когда Вы давали команду на сохранение Вашей Бейсик-программы. (Если бы Вы сохраняли коды, то такой же довесок был бы в самом последнем секторе файла <CODE>.)

Ведь это просто-напросто область ОЗУ, где набиралась Ваша командная строка, сохраненная вместе с Вашим файлом.

Вспомним, если Вы попытаетесь записать на диск всего один байт, то файл будет длиной в 1 сектор - 256 байт. Вот и в нашем случае программа была не кратна длине 256 и поэтому в файл записалась область, граничащая с нашей программой, а это как раз область редактора командной строки или EDIT SPACE.

Для эксперимента можете стереть всю область после байта #80 и, записав на диск, загрузить снова, дав команду LOAD...

Файл загрузится и никаких изменений в работе не произойдет. Получается, что этот довесок сектора пропадает?! Да, пропадает, и Вы можете использовать его по своему усмотрению - записав туда какую-нибудь информацию, (свое имя, например), а можете поместить там подпрограмму, например, дешифрации.

Правда, загружать эту подпрограмму из этого довеска Вам придется специальным загрузчиком. Команды TR-DOS этого сделать не смогут!

Правда, есть еще два байта в этом "довеске" (будем называть его потерянной областью файла), которые нужны системе - это байты АВТОЗАПУСКА Бейсика.

На рисунке 7 представлен дамп сектора этой же самой программы, но записанной с Функцией LINE 10, т.е. запуск со строки 10.

Track	25	Base:	5D3B	to	5D57	
00	0A	18	00	F5	22 50 52	U"PR
4F	47	52	41	4D	4D 20 4E	OGRAMM N
4F	20	41	55	54	4F 20 52	O AUTO R
55	4E	22	0D	80	AA 00 00	UN" *
35	36	31	39	0E	00 00 03	5619
3D	00	3A	EA	3A	F8 22 50	=:j:x"P
52	4F	47	32	22	CA 31 30	ROG1"J10
0D	80	CA	6B	5E	05 00 00	Jk↑
00	6B	5E	05	00	00 00 0A	k↑

Рис. 7 Дамп сектора программы бейсика с автостартом

На рисунке мы видим, что в системной области файла после кодовой последовательности #0D, #80, #AA, изменился 1 байт - вместо 00 стало #0A. Это и есть сигнал операционной системе, что после загрузки файл надо сразу запустить со строки 10. (Следующий байт тоже входит в сигнализатор запуска). Можете стереть (забить 00), все байты "довеска". Программа по команде LOAD самозапускаться не будет.

Инверсией в символьной части дампа показан "хвостик" командной строки, где видно, что после имени программы идет функция LINE 10; коды #CA, #31, 30.

Все вышеизложенное справедливо и для файлов типа <DATA >, с той лишь только разницей, что вместо тела программы будет находиться содержимое массива переменной (см. рис.8)

DATA	.D	Addr:	0	of	9	
46	49	4C	45	20	44 41 54	FILE DAT
41	50	AA	00	C1	35 36 31	A:*A561
39	0E	00	00	03	3D 00 3A	9:=":
EA	3A	F8	22	44	41 54 41	J:x"DATA
22	54	01	24	28	29 0D 80	"da\$(
F4	5B	5E	04	00	00 00 00	d [↑
00	00	00	00	00	00 00 00	k↑
6B	5E	05	00	00	00 6B 5E	k↑
05	00	00	00	0A	00 00 00	k↑

Рис. 8 Дамп сектора файла типа DATA

Для полноты изложения рассмотрим файлы с последовательным и параллельным доступом, тип #. При открытии файла и его записи на диск образуется файл длиной в 16 секторов, т.е. в 1 дорожку - независимо от того, ввели Вы в запись 1 байт информации или больше.

Правда, в заголовке файла, в каталоге, указывается величина в байтах, сколько Вы туда записали, но от этого не легче дорожка на диске уже занята. Оба типа файла записываются на диск одинаково, и если Вам придёт в голову последовательный файл открыть для чтения, как произвольный и наоборот - пожалуйста, система не обидится, но разбираться, какие записи, в каком порядке считывать, в этом случае, придется Вам самому.

С точки зрения программиста, который пишет программы на Ассемблере, все эти "довески" и нюансы не столь важны. В конечном итоге, ему нужно считать / записать информацию СЕКТОРА, (секторов), файла, а все остальное - дело техники.

Есть еще один вид файла, о котором пока не упоминалось в этой книге. Это, так называемые, "МАГИК ФАЙЛЫ", т.е. файлы, сброшенные ВОЛШЕБНОЙ КНОПКОЙ.

Но, поскольку этим файлам будет посвящена целая глава, то останавливаться здесь на этой теме мы не будем. Единственное, укажем, что эти файлы имеют тип<CODE> и всегда имеют длину файла, равную объему ОЗУ с адреса 16384 до 65535. Это слепок всего ОЗУ в файле, с сохраненными значениями всех регистров процессора.

Теперь, когда у нас есть все необходимые сведения о файлах, нам необходимо подробнее рассмотреть ту область ОЗУ, которую использует TR-DOS в своей работе, а именно, область системных переменных TR-DOS.

(Продолжение следует)

ЧИТАТЕЛЬ - ЧИТАТЕЛЮ



© Сергей Кучкин, г. Москва

MULTICOLOR - ЭФФЕКТЫ НА БОРДЮРЕ И КОЕ-ЧТО ЕЩЕ.

Как ни странно, эта тема на страницах журнала почти всегда обходилась стороной. Между тем, в последнее время MULTICOLOR встречается все чаще, а уж без бордюрных эффектов не обходится ни одна музыкальная демонстрация. Определимся с сутью вопроса:

Multicolor - расположение в одном знакоместе (8x8) более чем двух цветов (статическое либо динамическое);

Эффекты на бордюре - статические либо динамические изображения на бордюре.

Принцип, положенный в основу этих эффектов, довольно прост. Дело в том, что изменения в экранной памяти отображаются не мгновенно, а только при очередной прорисовке кадра (каждые 20 мс). Поэтому, быстро варьируя значениями атрибутов, можно добиться желаемого результата. Но тут начинаются проблемы...

1. Аппаратная несовместимость и проблемы синхронизации.

Чтобы хоть как-то компенсировать свою некомпетентность в аппаратных вопросах, я проштудировал все публикации ZX-РЕВЮ на тему INT'a, порта #FF и тому подобных вещей. Оставшиеся вопросы были проверены программно, но с одним ограничением: под рукой у меня есть ТОЛЬКО ПЕНТАГОН-128, и тестировал я ТОЛЬКО его.

1.1. Сплошные и раздельные поля памяти: конфликт Z80 и ULA. (Историческая справка).

При работе процессора, в блоке экранной памяти может возникнуть ситуация, когда запрос Z80 произойдет одновременно с запросом ULA (которая автоматически обновляет экран). В самых первых моделях Спектрума этот вопрос разрешался в пользу ULA. При этом на процессор переставали подаваться тактовые импульсы (другой метод состоял в подаче сигнала WAIT). Чем лучше была синхронизирована схема компьютера, тем реже возникали конфликты, и тем быстрее работал процессор.

Но, в любом случае, время исполнения одной и той же команды могло быть непостоянным, так что ни о какой полноценной синхронизации для создания тонких цветовых эффектов не могло быть и речи.

Если память компьютера организовывалась на одной микросхеме в 64К (речь пока идет о 48-ых Спектрумах), то конфликт возникал во всем адресном пространстве ("Москва - 48"). Если же память организовывалась на трех микросхемах по 16К, то конфликт возникал только в экранном блоке #4000-#8000 ("Пентагон-48?").

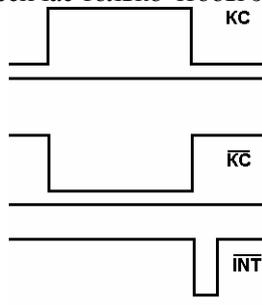
Известно, также, что в фирменных 128-ых моделях Спектрума страницы памяти 0-3 были бесконфликтными (не состязательными), а 4-7 конфликтными (состязательными).

Позднее появились машины с раздельными полями памяти ("Пентагон-128"). В них для ULA отвели отдельное Видео-ОЗУ, и она перестала тормозить процессор. Программы одинаково быстро заработали во всем адресном пространстве, что гораздо упростило синхронизацию при работе с экраном.

1.2. Экран: 312 против 320. Синхронизация по INT'у.

Сигнал на перерисовку экрана - кадровый синхроимпульс вырабатывается каждые 20 миллисекунд. По его заднему фронту идет INT (сигнал прерывания) (тут я сошлюсь на Кирилла Громова, ZX-РЕВЮ 95/3).

Длительность INT'a по классическим меркам должна составлять 8-9 мкс, то есть 28-32 такта. Но это теория. На практике же начало INT'a может и не совпадать с концом синхроимпульса, а длительность кое-где достигает 15 мкс (52-53 такта). Главное для нас сейчас только чтобы эти величины были постоянными.



Прорисовка экрана начинается с самого верха бордюра и заканчивается в самом его низу. В моем Пентагоне экран состоит из 320 строк по 224 такта на каждую строку. Таким образом, всего получается $320 \times 224 = 71680$ тактов/экран (очевидно, столько же тактов/INT). Однако, существует другой вариант, считающийся более правильным (?). Он реализован, например, в "ПРОФИ". Это 312 строк в экране. Если сделать логичное предположение, что одна строка прорисовывается также за 224 такта, то получим $312 \times 224 = 69888$ тактов/INT. Если Вы захотите проверить число строк на экране своего компьютера, то можете воспользоваться

предложенной ниже программой LINETEST. После вызова из Бейсика PRINT USR ... она вернет количество строк в паре BC (исходя из 224 тактов в строке). У меня этот тест дал, как и ожидалось, 320. Проверить его на "Профи", к сожалению, не было возможности.

```

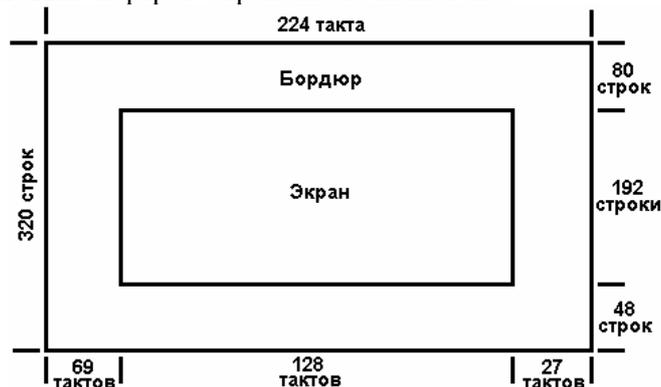
; LINETEST 312/320/...?
    ORG    #8000
    ENT    $
    DI                    ; Запрещаем прерывания.
    LD     HL, INT        ; Останавливаем адрес программы обработки
    LD     (#80FF), HL    ; прерываний на процедуру INT.
    LD     A, #80
    LD     I, A
    IM     2
    LD     DE, 0          ; Обнуляем счетчик строк.
    EI                    ; Разрешаем прерывания.
    HALT                   ; Ждем очередного синхроимпульса.
    LD     HL, INT2       ; Переопределяем обработку прерываний
    LD     (#80FF), HL    ; на процедуру INT2, которая остановит выполнение
                                ; после прихода следующего синхроимпульса.
M0      INC     DE        ; Увеличиваем счетчик строк.
        LD     B, 15      ; Пропускаем необходимые такты.
M1      DJNZ   M1
        LD     A, R        ; !!! команда для пропуска еще 9 тактов!
        JR     M0          ; Цикл замкнулся.
INT     EI
        RET
INT2    POP     HL        ; Снимаем со стека адрес возврата.
        LD     C, E        ; Полученное число строк
        LD     B, D        ; пересылаем в BC.
        IM     0          ; возвращаемся
        EI                    ; в
        RET                ; Бейсик.

```

Обратите внимание! Здесь и далее комментарием «!!!» у меня обозначены команды, выполняемые вхолостую (только ради тактов).

Стоит заметить, что при выполнении процедуры INT, прерывание может захватываться повторно (даже при нормальной длине сигнала INT), но это не страшно, так как лишние 30-40 тактов не способны повлиять на результат теста. (Потенциально возможны ситуации, когда в экране умещается нецелое число строк, например, 312.5; в подобных случаях тест ошибется, если, конечно, не произвести в нем соответствующих изменений).

Теперь я приведу тактовый "портрет" экрана моего Пентагона.



Отсюда видно, например, что каждая строка "маленького" экрана (256x192) прорисовывается отдельно. А вот пиксели в строке рисуются по два за такт (256 пикселей за 128 тактов).

Можно заметить, что верхнему левому углу "малого" экрана соответствует 69-й такт 80 строки. Эти величины как раз и характеризуют положение INT'а относительно кадрового синхроимпульса. Не исключено, что они колеблются даже у компьютеров одного типа (однако это не факт). Кстати, эти 69 тактов долго не давали мне покоя (число ведь не круглое!). Я перемерял так и сяк, но результат был один и тот же. В конце концов, я решил, что это связано с длиной кадрового синхроимпульса (ведь INT идет позже начала перерисовки экрана как раз на этот интервал времени, а я отсчитывал такты, разумеется, от INT'а).

Как же синхронизировать программу с перерисовкой экрана? Например, просто HALT'ом:

```

...
LD     HL, INT        ; INT - адрес программы обработки

```

```

LD      (#XXFF) , HL      ; прерываний.
LD      A, #XX
LD      I, A
IM      2
EI
HALT

```

INT

Вот и все, программа придет на INT почти сразу после начала перерисовки экрана. Теперь выводите, например, в порт #FE каждые 71680 (69888) тактов что-нибудь, и у Вас готова картинка на бордюре. Вставив небольшую первоначальную задержку легко разместить бордюрную картинку там, где нужно. При желании, можно даже заставить ее двигаться (варьируя число тактов между перерисовками). В «Приложении Б» приведен пример программы CURSOR, которая делает нечто подобное плюс еще кое-что.

1.3. Синхронизация с помощью порта #FF: пара цитат.

Порт #FF! По своей популярности он давно уже побил все рекорды. Корреспонденты приходили и уходили, а порт #FF успешно просуществовал на страницах ZX-РЕВЮ аж с 1991 года и до наших дней! Причем редкий номер не обходился без нового "окончательного и бесповоротного" решения проблемы. Так в чем же суть?

В фирменном Спектруме при чтении из порта #FF (как впрочем, и из любого другого несуществующего порта) считываются атрибуты, выводимые в данный момент на экран, попеременно с кодом #FF. Почему это происходит, объяснили уже давно (ZX-РЕВЮ за 1991 год).

Многие отечественные компьютеры (если не сказать, подавляющее большинство) не имеют порта #FF (то есть, из него ВСЕГДА читается код #FF). И не было бы много печали, если бы некоторые программы не использовали метод синхронизации с помощью этого порта. Его суть вот в чем:

```

WAIT    IN      A, (#FF)      ; Этот цикл выполняется до тех пор,
INC     A        ; пока в порту не появится байт атрибутов
JR      Z, WAIT   ; (то есть началась перерисовка экрана) .

```

Очевидно, что при отсутствии порта #FF программа просто повиснет.

Существуют тесты, позволяющие проверить наличие этого порта у Вас (у меня, например, порта нет, что и заставляет меня прекратить рассказ, не подтвержденный экспериментами).

1.4. INT и его повадки: важные мелочи.

До тех пор, пока Вы работаете отдельно с экраном, отдельно с бордюром, всего вышесказанного должно быть достаточно. Но если Вы захотите синхронизировать экранное изображение с бордюрным, то тут появится проблема.

Запустив мою программу CURSOR, Вы должны были заметить, что "бордюрная" часть курсора постоянно либо "убегает", либо отстает от "экранной" (поэтому и была введена клавиша '0'). Такое поведение курсора объясняется не просчетом в программе, а свойством системы прерываний.

Дело в том, что когда приходит сигнал INT, процессор может быть занят выполнением команды. В таком случае он сначала завершит команду, а только потом перейдет по прерыванию. Задержка может составлять от 1 до 20-26 (?) тактов в зависимости от выполняемой команды. Между тем, прорисовка экрана уже идет полным ходом, то есть переход по INT запаздывает. В случае с HALT'ом (который кто-то назвал NOP'ом без увеличения PC), задержка может составлять 0, 1, 2 или 3 такта. Если вспомнить, что за один такт прорисовываются два пикселя экрана, то за 3 такта: $2 \times 3 = 6$ - почти целое знакоместо!

Возникает вопрос, можно ли избавиться от этого эффекта программно? Ответ: можно. Идея реализована в приведенной ниже программе FILTER. Она использует ячейки #81FF, #8200, изменяет I и некоторые другие регистры. После вызова CALL FILTER сигнал INT соответствует началу следующей команды (но прерывания запрещены).

```

; INT FILTER ONLY FOR PENTAGON-128
ORG     #8000
ENT     $
FILTER  DI
LD      HL, INT      ; Останавливаем процедуру обработки
LD      (#81FF) , HL ; прерываний INT.
LD      A, 0         ; Очищаем дно буфера.
LD      (BUFF+4) , A
LD      E, 1         ; Счетчик установлен в 1.
LD      HL, BUFF     ; HL - указатель на первый свободный
LD      A, #81       ; байт в буфере.
LD      I, A
IM      2

```

```

EI
HALT ; Ждем прерывание. Теперь у нас задержка INTа 0,
; 1, 2 или 3 такта, а в (HL)=1 в любом случае.
CALL WAIT ; Задержкана71677тактов. Таким образом перед
; прерыванием успеет начать выполняться INC E для
; задержек 0, 1 и 2, а для 3 уже нет. В
; результате: (HL)=2 для случаев 0, 1 и 2 и
; (HL)=1 для случая 3.
INC E ; 2
CALL WAIT ; Аналогичный "отсев" по одному.
INC E ; 3
CALL WAIT ;
INC E ; 4
DI ; Запрещаем прерывания. Теперь расклад такой (см.
; далее таблицу):

```

Случай	BUFF+0	BUFF+1	BUFF+2	BUFF+3
0	1	2	3	4
1	1	2	3	3
2	1	2	2	3
3	1	1	2	3

```

; Осталось проверить ячейки и сделать
; соответствующие задержки.
LD BC,2749 ; В BC - число повторений в цикле задержки
; ((BC-1)x26+21 тактов).
LD HL,(BUFF+1)
EX DE,HL
LD HL,(BUFF+3)
LD A,H
OR A
JR NZ,F2 ; Если в (BUFF+4) не ноль, то произошел
; гипотетический случай: прерывание" вклинилось"
; между EI и HALT. (См. начало программы). Это
; приведет лишь только к сдвигу данных на один
; байт, что и устраняем).
F0 LD A,L
CP 4
JR Z,DISP3 ; Начальное смещение 0!
LD A,D
CP L
JR Z,DISP2 ; Начальное смещение 1!
LD A,E
CP D
JR Z,DISP1 ; Начальное смещение 2!
DISP0 NOP ; !!! Обработка смещения 3.
LD R,A ; !!!
LD R,A ; !!!
F1 DEC BC ; Ждем рассчитанное число тактов до
LD A,B ; следующего INT'а
OR C
JR NZ,F1
RET ; Возврат.
F2 LD E,D ; В случае, если прерывание "вклинилось" между
LD D,L ; EI и HALT (в начале),
LD L,H ; надо сдвинуть все на 1 байт.
DEC BC ; Уменьшаем счетчик задержки, компенсируя время,
DEC BC ; потраченное на перезагрузки
LD A,(HL) ; !!!
NOP ; !!!
JR F0
DISP1 DEC BC ; Обработка смещения 2.
NOP ; !!!
JR DISP0
DISP2 JR DISP0 ; Обработка смещения 1.

```

```

DISP3   LD     R,A           ; !!! Обработка смещения 0.
        LD     A,(HL)       ; !!!
        JR     DISPO
WAIT    LD     BC,2753      ; Задержка на 71617 тактов
W1      DEC    BC           ; (включая время на вызов CALL WAIT).
        LD     A,B
        OR     C
        JR     NZ,W1
        LD     B,(HL)       ; !!!
        RET
INT     LD     (HL),E
        INC    HL
        LD     D,(HL)       ; !!!
        LD     D,(HL)       ; !!!
        EI
        RET
BUFF    DEFS   5           ; Буфер для хранения результатов.

```

Процедура довольно "сырая" (как, впрочем, и все программки в этой статье).

В принципе, ее несложно переработать и для "Профи" (69888 тактов/INT), но я не стал этого делать из-за невозможности отладки на "Пентагоне".

Напомню, что комментариями отмечались команды, не несущие смысловой нагрузки, а только забирающие время.

Программу CURSOR, модернизированную с помощью "фильтра", Вы найдете в «Приложении Б» под названием SCURS. Теперь ничто ни от чего не отстаёт, все в порядке (за исключением мелочи: бордюрное изображение у меня всегда отстает от экранного примерно на 1 пиксель, то есть 0.5 такта, что, очевидно, программно исправить - невозможно).

Правда, у FILTER'a есть недостаток - он долго работает (несколько прерываний). Таким образом, с его помощью возможна только первоначальная синхронизация, а когда кадры пропускать нельзя, Вам придется выкручиваться без него.

2. MULTICOLOR.

Может сложиться впечатление, что multicolor - вещь редкостная, и, ввиду своей сложности, является делом музыкальных демонстраций и заставок. Я попытаюсь доказать, что это не так.

Что же нужно для multicolor'a?

- a) Синхронизироваться с началом перерисовки экрана;
- b) Дождаться перерисовки атрибутов (то есть пропустить, как минимум, первые 80 строк);
- c) Дать компьютеру прорисовать первую "полоску" атрибутов, но прежде, чем он доберется до второй, заменить атрибуты на другие.

Таким образом, в одном знакоместе можно разместить до $2 \times 8 = 16$ цветов, то есть, все цвета Спектрума! Цвет задается отдельно для каждого байта знакоместа. Осталось только повторять подобный трюк в каждом кадре, и у Вас на экране полноценный multicolor!

Кстати, для обладателей "старого доброго" RESET'a могу порекомендовать надежный тест. Если Вы сомневаетесь: multicolor перед Вами или обман, то жмите RESET, но не отпускайте его некоторое время. При этом multicolor "рассыплется", а обычная картинка будет преспокойно висеть на экране.

Возможно, у Вас возникнет желание перерисовывать атрибуты настолько быстро, чтобы получился multicolor в пределах одного байта знакоместа. Но хочу Вас огорчить - этот трюк не удастся. Горизонтальный multicolor невозможен, и вот почему.

Байт из экранной памяти вместе со своим атрибутом перед выводом на экран попадает в специальный сдвиговый регистр, недоступный программно. Между считыванием байта из памяти и засылкой его на экран - интервал в 3 такта. Таким образом, первый байт экрана считывается на 66-м такте 80-й строки (нумеруя с 0). Причем любые изменения памяти в промежутке между считыванием и изображением станут заметны только в следующем кадре.

Но вернемся к multicolor'у вертикальному. Тут не все так мрачно, хотя есть и свои проблемы. Небольшой расчет: прорисовка строки экрана занимает 224 такта; переброска 1 байта с помощью LDI (LDD) занимает 16 тактов; $224/16=14$, то есть, максимум 14 знакомест с multicolor'ом в строке (это даже без учета установок для LDI, а с ними и того меньше). Что же делать?

Как всегда, выручает стек. Придется, также, отбросить очень медленные команды перехода, то есть развернуть все циклы. У нас получится "рыхлый" код размером в несколько килобайт, но игра стоит свеч.

Есть две проблемы: каким способом брать атрибуты из памяти и каким класть их на экран (трудно нагрузить и то и другое на один стек). Однозначного ответа быть не может. Если Вас устраивает жесткая фиксация области multicolor'a на экране, то Вы немного выигрываете в скорости:

```
LD      SP, #XXXX      ; #XXXX - адрес в памяти
.....
POP     HL
LD      (#YYYY), HL
POP     HL
LD      (#ZZZZ), HL    ; #YYYY, #ZZZZ адреса на экране
.....
```

и так далее для всей области. Замечу, что POP быстрее, чем PUSH на 1 такт; а LD (#ADDR),HL быстрее LD (#ADDR),DE на 4 такта. При такой процедуре вывода достаточно изменить атрибут в памяти, и при следующей прорисовке он изобразится на экране. Теперь подсчет: "POP HL" + "LD (#YYYY),HL" = 26 тактов; $224=8 \times 26 + 16$. То есть, получаем $2 \times 8 = 16$ знакомест (каждый "POP HL" даёт два), плюс еще 16 холостых тактов.

Следующий метод позволяет более гибко менять положение области multicolor'a на экране:

```
LD      SP, #ADDR      ; #ADDR - адрес на экране
LD      HL, (#XXXX)
PUSH   HL
LD      HL, (#YYYY)    ; #XXXX, #YYYY адреса в памяти
PUSH   HL
.....
```

и так далее для всей области. Обратите внимание, что в отличие от предыдущего случая, где могла быть только одна команда загрузки SP, здесь такая команда нужна при выводе каждой строчки (из-за причудливости спектрумовского экрана). Зато для изменения местоположения multicolor'a достаточно изменить "только" все #ADDR. Это немало, но сравните с предыдущим случаем. Теперь расчет: "LD HL, (#XXXX)" + "PUSH HL" = 27 тактов и еще 10 тактов выполняется "LD SP, #ADDR", итого: $224 - 10 = 214 = 7 \times 27 + 25$. То есть, получаем $2 \times 7 = 14$ знакомест в строке и 25 холостых тактов (право, обидно!).

Может возникнуть резонный вопрос: "А нельзя ли косвенно адресовать экран, то есть, вместо LD SP, #ADDR написать LD SP, (#ADDR)?" Ответ: «Нельзя». То есть, конечно, можно, но придется пожертвовать еще парой знакомест. Дело в том, что направление прорисовки экрана и направление заполнения стека противоположны. Вследствие этого, во избежание простоя, мы должны перегружать SP не один, а два раза за строку.

Подводя итог вышесказанному, осталось заметить, что алгоритмы, аналогичные вышерассмотренным реализованы в программах MAXSCM и 16X16MC. Первая представляет собой реализацию multicolor'a для любого рисунка размером 18×14 знакомест. С моей точки, зрения это максимально широкий по X multicolor, реализуемый для произвольных атрибутов (в специфических случаях, безусловно, его можно расширить). По Y размер картинки не критичен, и может, в принципе, достигать' всего экрана. 16X16MC реализует multicolor в квадрате 16×16 по первому из рассмотренных алгоритмов. Обе программы сначала создают сам работающий блок, а потом его запускают. При этом на экране может быть любое изображение, а файл атрибутов multicolor'a должен находиться по адресу 49152. Чуть подробнее программы описаны в «Приложении Б».

После прочтения предыдущих абзацев у Вас, должно быть, сложилось впечатление, что все свое время процессор должен посвятить multicolor'у, забросив остальные дела. Но это не так! Рассмотрим, например, multicolor 16×16 (по площади 1/3 экрана). На каждый кадр приходится 42956 тактов "безделья", что ни много, ни мало, а почти 3/5 всего времени процессора. Его можно занять, например, полным вертикальным scroll'ом этого самого квадрата 16×16 с точностью до точки, пустить текст нестандартным шрифтом со скоростью 5 строк в секунду (больше, чем достаточно), и еще время останется!

Как видно из всего вышесказанного, multicolor открывает поистине необозримые возможности для создания мелкой цветной графики. Остается только удивляться, что они не использованы до сих пор. Кстати, второй экран 128 компьютера мне так и не удалось задействовать для улучшения работы multicolor'a. Вот если бы разрешалось фиксировать экран на любой из 8 страниц... Тогда ПОЛНОЭКРАННЫЙ multicolor сводился бы к простой цепочке OUT'ов в порт #FD (несмотря на то, что этот OUT делает очень многое, он выполняется не дольше обычного).

Быть может, широкому распространению multicolor'a мешает аппаратная несовместимость? Но этот момент преодолим. Сейчас я разрабатываю систему multicolor'a, автоматически распознающую компьютер, и подстраивающуюся под него. Но тут катастрофически не хватает информации по распространенным синхросистемам. Если кто-нибудь заинтересуется подробным вопросом, то мой адрес и телефон даны ниже. Пишите, звоните!

3. Вместо послесловия.

Кучкин Сергей

121096, г. Москва, ул. Минская, д.14, кв.102; тел. 144-26-00

Приложение А.

Программируя multicolor, я столкнулся с одной проблемой: все имеющиеся у меня готовые процедуры теперь не годились, поскольку время их выполнения зависело от начальных условий. Так, например, при написании scroller'a под multicolor'ом мне пришлось заново переписывать вывод нестандартным шрифтом, опрос клавиатуры и все подобные мелочи.

Необходимость точно согласовывать время выполнения программы при любых условиях, порождает особый стиль программирования. Погоня за скоростью и малым размером отступает на второй план.

Приходится минимизировать число условных переходов и вызовов, которые сильно осложняют ситуацию.

Ниже я рассмотрю несколько способов, позволяющих сделать время выполнения программы независимым от начальных условий.

Способ 1: прямая коррекция времени.

Каждый условный переход должен превратиться в подобную конструкцию:

```
.....
JP      MC, M1
NOP                                ; !!! Тут NOP и .. обозначают команды, выполняемые
.....                                ; для выравнивания
JP      M2                          ; времени в случае соблюдения условия
M1      NOP                          ; !!! и в случае его несоблюдения.
.....
```

Этот способ хорош при небольшом числе необходимых переходов. Но вот с CALL'ами он справиться уже не может. Тут применим способ 2.

Способ 2: косвенная коррекция времени.

Один из регистров (регистровых пар) или ячейка памяти отводится под счетчик задержки в конце программы. Вначале в этот счетчик заносится время задержки, характерное для самой быстрой работы (не происходило условных переходов и заходов в CALL'ы). При вызове каждая из подпрограмм корректирует этот счетчик в зависимости от времени, необходимого для ее выполнения. Поскольку единица счетчика, скорее всего, соответствует нескольким десяткам тактов, то тут не обойтись, также, без прямой коррекции. Но хватит слов. Вот пример:

```
LD      BC, #300                    ; Инициализируем счетчик.
.....
CALL    NC, M1
.....
CALL    Z, M2
M3      DEC    BC                    ; Задержка в соответствии со счётчиком
LD      A, B                          ; (единица счетчика равна 26 тактам) .
OR      C
JR      NZ, M3
RET
M1      DEC    BC                    ; "Занимаем" у счетчика 52 такта,
DEC    BC                              ; необходимые для работы.
NOP                                ; !!! "Возвращаем" 4 лишних.
.....                                ; Продолжаем
```

Однако, бывают случаи (особенно при адаптации готовых процедур), когда вносить изменения в структуру программы крайне нежелательно. Тогда применим самый трудоемкий способ:

Способ 3: способ "супервизора".

Когда нам необходимо вызвать процедуру, которая не заботится о времени своего выполнения, мы вызываем не ее, а специально подготовленную программу-супервизор. Эта программа анализирует переданные параметры, делает соответствующие задержки, и только потом вызывает процедуру.

Метод плох тем, что необходимо для каждого случая рассчитать число тактов выполнения, заложить эту информацию в "супервизор". И не удивительно, если он после этого получится больше самой программы. Что поделаешь, издержки неизбежны!

А теперь небольшой список команд, которыми можно забивать время в несколько тактов, если в этом возникнет необходимость:

4 такта	NOP; LD reg8,reg8
5 тактов	RET <условие> (если условие заведомо ложно!)
6 тактов	INC reg16; DEC reg 16
7 тактов	LD reg8,(HL); CP #00; ADD A,#00
8 тактов	NEG; IM 0/1/2

9 тактов	LD R,A
10 тактов	LD reg16,#NNNN; INC IX; DEC IX
11 тактов	INC (HL); ADD HL,reg16; OUT (#NN),A
12 тактов	IN reg8,(C); OUT (C),reg8
13 тактов	LD A,(#NNNN)
14 тактов	LD IX,#NNNN
15 тактов	PUSH IX; ADD IX,reg16; ADC HL,reg16
16 тактов	CPI; CPD; LD HL,(#NNNN)
17 тактов	CALL #NNNN
18 тактов	RLD; RRD
19 тактов	LD reg8,(IX+d); CP (IX+d)
20 тактов	LD reg16,(#NNNN)
21 такт	Некоторые команды при специфических условиях
23 такта	RLC (IX+d); SET 0,(IX+d); EX (SP),IX

Обозначения:

reg8 - A,B,C,D,E,H,L (для IN, F);

reg16 - BC, DE, HL и AF либо SP (что допустимо);

#NN - байт;

#NNNN - слово;

d - смещение в командах с индексными регистрами.

IX везде свободно заменяется на IY.

Это, разумеется, далеко не полный список. Некоторые команды из приведенных выше предпочтительнее по размеру, другие - по влиянию на флаги.

Отдельно следует заметить, что переход по маскируемому прерыванию занимает 19 тактов (без учета возможной задержки, то есть в случае, когда прерывание пришло перед началом выполнения команды).

Приложение Б.

Первая программа LINETEST - возвращает в регистровой паре BC число строк в экранном кадре компьютера (если в строке по 224 такта). Возможен вызов из Бейсика по PRINT USR

```
;LINETEST 312/320/...?
```

```

    ORG    #8000
    ENT    $
    DI
    LD     HL, INT
    LD     (#80FF), HL
    LD     A, #80
    LD     I, A
    IM     2
    LD     DE, 0
    EI
    HALT
    LD     HL, INT2
    LD     (#80FF), HL
M0      INC     DE
    LD     B, 15
M1      DJNZ   M1
    LD     A, R           ; !!!
    JR     M0
INT     EI
    RET
INT2    POP    HL
    LD     C, E
    LD     B, D
    IM     0
    EI
    RET

```

<p>ВСЕ ОСТАЛЬНЫЕ ПРОГРАММЫ СИНХРОНИЗИРОВАНЫ ТОЛЬКО ПОД "ПЕТАГОН"!</p>

Программа CURSOR позволяет перемещать курсор 3x3 знакоместа, как по экрану, так и по бордюру. В силу изложенных ранее причин курсор на бордюре может отставать от экранного. Ситуацию можно исправить, нажимая клавишу '0'. Управление синклер-джостиком, выход SPACE.

```

;CURSOR
    ORG      #8000
    ENT     $
    DI
    EXX
    PUSH    HL
    LD      HL,INT
    LD      (#81FF),HL
    LD      HL,#0509
    LD      (X),HL
    LD      BC,#2FF
    LD      DE,#5801
    LD      HL,#5800
    LD      (HL),#0F
    LDIR
    LD      A,#81
    LD      I,A
    IM      2
    EI
    HALT
    DI
    LD      BC,473
M0      DEC    BC
    LD      A,B
    OR      C
    JR      NZ,M0
    INC     BC          ; !!!
    INC     BC          ; !!!
    LD      R,A        ; !!!
M1      LD      C,#FE
    LD      HL,#0107
    CALL    DRAW
    LD      BC,2334
    CALL    INKEY
    CALL    WAIT
    LD      DE,0        ; !!!
    LD      DE,0        ; !!!
    LD      D,0         ; !!!
    LD      BC,#7FFE
    IN      D,(C)
    RR      D
    JR      C,M1
    LD      BC,#2FF
    LD      DE,#5801
    LD      HL,#5800
    LD      (HL),#0F
    LDIR
    POP     HL
    EXX
    IM      0
    EI
    RET
INT      EI
    RET
X        DEFB    #09
Y        DEFB    #05
DRAW     LD      E,#18
D1       LD      B,14
D2       DJNZ   D2
    OUT     (C),L

```

```

        OUT      (C),H
        DEC      E
        JR       NZ,D1
        RET
WAIT    DEC      BC
        LD       A,B
        OR       C
        JR       NZ,WAITRET
INKEY   LD       A,#0F
        CALL    DRAW2
        LD       A,#EF
        IN      A, (#FE)
        RRA
        CALL    NC,FIRE
        RRA
        CALL    NC,UP
        RRA
        CALL    NC,DOWN
        RRA
        CALL    N,RIGHT
        RRA
        CALL    N,LEFT
        LD       A,#38
        CALL    DRAW2
        RET
FIRE    LD       HL,-2
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      D,0          ; !!!
        RET
UP      LD       HL,-74
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(Y)
        DEC     A
        CP      1
        LD      (Y),A
        LD      A,E
        CALL    Z,DOWN
        LD      DE,0          ; !!!
        LD      DE,0          ; !!!
        LD      D,0          ; !!!
        NOP     ; !!!
        RET
DOWN    LD       HL,64
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(Y)
        INC     A
        CP      #24
        LD      (Y),A
        LD      A,E
        CALL    Z,UP
        LD      R,A          ; !!!
        LD      R,A          ; !!!
        LD      R,A          ; !!!
        RET
RIGHT   LD       HL,#FFFC
        ADD     HL,BC

```

```

LD      C,L
LD      B,H
LD      E,A
LD      A,(X)
INC     A
CP      #2F
LD      (X),A
LD      A,E
CALL    Z,LEFT
LD      E,0          ; !!!
RET
LEFT   LD      HL,#FFFB
ADD     HL,BC
LD      C,L
LD      B,H
LD      E,A
LD      A,(X)
DEC     A
CP      6
LD      (X),A
LD      A,E
CALL    Z,RIGHT
LD      R,A          ; !!!
LD      R,A          ; !!!
LD      D,0          ; !!!
RET
DRAW2  EX      AF,AF
PUSH    BC
LD      DE,(X)
R1     LD      C,3
LD      B,3
LD      H,0
LD      A,E
CP      12
RL      H
CP      44
CCF
RL      H
R2     LD      A,D
LD      1,0
CP      8
RL      L
CP      32
CCF
RL      L
EXX
LD      C,A
EX      AF,AF
LD      B,A
LD      A,C
SUB     8
LD      L,A
LD      H,0
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
EXX
LD      A,E
EXX
SUB     12
LD      E,A
LD      D,0

```

```

ADD    HL,DE
LD     DE,#5800
ADD    HL,DE
EXX
LD     A,H
OR     L
EXX
JR     NZ,R4
LD     DE,0           ; !!!
LD     (HL),B
R3    LD     A,B
EX     AF,AF'
EXX
INC    A
DJNZ   R2
INC    E
DEC    C
JR     NZ,R1
POP    BC
RET
R4    JR     R3

```

Программа - FILTER. Компенсирует произвольную задержку INT'a при ожидании исполнения команды. После ее вызова прерывания запрещены, а сигнал INT соответствует началу следующей команды. Портится содержимое I, и других регистров.

```

;INT FILTER
ORG    #8000
ENT    $
FILTER DI
LD     HL,INT
LD     (#81FF),HL
LD     A,0
LD     (BUFF+4),A
LD     E,1
LD     HL,BUFF
LD     A,#81
LD     I,A
IM     2
EI
HALT
CALL   WAIT
INC    E           ; 2
CALL   WAIT
INC    E           ; 3
CALL   WAIT
INC    E           ; 4
DI
LD     BC,2749
LD     H1,(BUFF+1)
EX     DE,HL
LD     HL,(BUFF+3)
LD     A,H
OR     A
JR     NZ,F2
F0    LD     A,L
CP     4
JR     Z,DISP3
LD     A,D
CP     L
JR     Z,DISP2
LD     A,E
CP     D
JR     Z,DISP1
DISP0 NOP           ; !!!

```

```

LD      R,A          ; !!!
LD      R,A          ; !!!
F1      DEC          BC
LD      A,B
OR      C
JR      NZ,F1
RET
F2      LD          E,D
LD      D,L
LD      L,H
DEC     BC
DEC     BC
LD      A,(HL)      ; !!!
NOP     ; !!!
JR      F0
DISP1   DEC          BC
NOP     ; !!!
JR      DISP0
DISP2   JR          DISP0
DISP3   LD          R,A      ; !!!
LD      A,(HL)      ; !!!
JR      DISP0
WAIT    LD          BC,2753
W1      DEC          BC
LD      A,B
OR      C
JR      NZ,W1
LD      B,(HL)      ; !!!
RET
INT     LD          (HL),E
INC     HL
LD      D,(HL)      ; !!!
LD      D,(HL)      ; !!!
EI
RET
BUFF   DEFS        5

```

Программа CURSOR, оснащенная фильтром INT'a. Управление синклер-джостиком, SPACE - выход.

Возможно отставание бордюрного курсора на один пиксель (что неустранимо программно).

; SCURS

```

ORG     #8000
ENT     $
JP      CURSOR
FILTER  DI
LD      HL,INT
LD      (#81FF),HL
LD      A,0
LD      (BUFF+4),A
LD      E,1
LD      HL,BUFF
LD      A,#81
LD      I,A
IM      2
EI
HALT
CALL    WAIT
INC     E          ;2
CALL    WAIT
INC     E          ;3
CALL    WAIT
INC     E          ;4
DI
LD      B,2749
LD      HL,(BUFF+1)
EX      DE,HL

```

```

LD      HL, (BUFF+3)
LD      A, H
OR      A
F0      JR      NZ, F2
LD      A, L
CP      4
JR      Z, DISP3
LD      A, D
CP      L
JR      Z, DISP2
LD      A, E
CP      D
JR      Z, DISP1
DISP0   NOP      ; !!!
LD      R, A      ; !!!
LD      R, A      ; !!!
F1      DEC      BC
LD      A, B
OR      C
JR      NZ, F1
RET
F2      LD      E, D
LD      D, L
LD      L, H
DEC     BC
DEC     BC
LD      A, (HL)   ; !!!
NOP     ; !!!
JR      FO
DISP1   DEC     BC
NOP     ; !!!
JR      DISP0
DISP2   JR      DISP0
DISP3   LD      R, A      ; !!!
LD      A, (HL)   ; !!!
JR      DISP0
WAIT    LD      BC, 2753
W1      DEC     BC
LD      A, B
OR      C
JR      NZ, W1
LD      B, (HL)   ; !!!
RET
INT     LD      (HL), E
INC     HL
LD      D, (HL)   ; !!!
LD      D, (HL)   ; !!!
EI
RET
BUFF    DEFS    5
CURSOR  DI
EXX
PUSH    HL
LD      HL, #0509
LD      (X), HL
LD      BC, #2FF
LD      DE, #5801
LD      HL, #5800
LD      (HL), #0F
LDIR
CALL    FILTER
LD      BC, 475
M0      DEC     BC
LD      A, B

```

```

OR      C
JR      NZ,M0
LD      R,A          ; !!!
M1      LD      C,#FE
LD      HL,#0107
CALL    DRAW
LD      BC,2334
CALL    INKEY
CALL    WAIT2
LD      DE,0        ; !!!
LD      DE,0        ; !!!
LD      D,0         ; !!!
LD      BC,#7FFE
IN      D,(C)
RR      D
JR      C,M1
LD      BC,#2FF
LD      DE,#5801
LD      HL,#5800
LD      (HL),#0F
LDIR
POP     HL
EXX
IM      0
EI
RET
X      DEFB      #09
Y      DEFB      #05
DRAW   LD      E,#18
D1     LD      B,14
D2     DJNZ    D2
OUT    (C),L
OUT    (C),H
DEC    E
JR     NZ,D1
RET
WAIT2  DEC    BC
LD     A,B
OR     C
JR     NZ,WAIT2
RET
INKEY  LD     A,#0F
CALL   DRAW2
LD     A,#EF
IN     A,(#FE)
RRA
CALL   NC,FIRE
RRA
CALL   NC,UP
RRA
CALL   NC,DOWN
RRA
CALL   NC,RIGHT
RRA
CALL   NC,LEFT
LD     A,#38
CALL   DRAW2
RET
FIRE   LD     HL,-2
ADD    HL,BC
LD     C,L
LD     B,H
LD     D,0      ; !!!
RET

```

```

UP      LD      HL,-74
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(Y)
        DEC     A
        CP      1
        LD      (Y),A
        LD      A,E
        CALL    Z,DOWN
        LD      DE,0      ; !!!
        LD      DE,0      ; !!!
        LD      D,0      ; !!!
        NOP     ; !!!
        RET

DOWN    LD      HL,64
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(Y)
        INC     A
        CP      #24
        LD      (Y),A
        LD      A,E
        CALL    Z,UP
        LD      R,A      ; !!!
        LD      R,A      ; !!!
        LD      R,A      ; !!!
        RET

RIGHT   LD      HL,#FFFC
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(X)
        INC     A
        CP      #2F
        LD      (X),A
        LD      A,E
        CALL    Z,LEFT
        LD      E,0      ; !!!
        RET

LEFT    LD      HL,#FFFB
        ADD     HL,BC
        LD      C,L
        LD      B,H
        LD      E,A
        LD      A,(X)
        DEC     A
        CP      6
        LD      (X),A
        LD      A,E
        CALL    Z,RIGHT
        LD      R,A      ; !!!
        LD      R,A      ; !!!
        LD      D,0      ; !!!
        RET

DRAW2   EX      AF,AF
        PUSH    BC
        LD      DE,(X)
        LD      C,3
R1      LD      B,3

```

```

LD      H, 0
LD      A, E
CP      12
RL      H
CP      44
CCF
RL      H
LD      A, D
R2      LD      L, 0
CP      8
RL      L
CP      32
CCF
RL      L
EXX
LD      C, A
EX      AF, AF'
LD      B, A
LD      A, C
SUB     8
LD      L, A
LD      H, 0
ADD     HL, HL
EXX
LD      A, E
EXX
SUB     12
LD      E, A
LD      D, 0
ADD     HL, DE
LD      DE, #5800
ADD     HL, DE
EXX
LD      A, H
OR      L
EXX
JR      NZ, R4
LD      DE, 0          ; !!!
LD      (HL), B
R3      LD      A, B
EX      AF, AF
EXX
INC     A
DJNZ   R2
INC     E
DEC     C
JR      NZ, R1
POP     BC
RET
R4      JR      R3

```

Программа для создания статического multicolor'a в отцентрованном прямоугольнике 18x14 знакомест. Изображение должно уже находиться на экране, а атрибуты - с адреса SCR=#C000 в формате 18 байт на каждую строку x 112 строк (по порядку).

```

;MAXIMUM STATIC MULTICOLOR (X)
ORG     #8000
SCR     EQU     #C000
ENT     $
LD      DE, L4
LD      IX, SCR

```

```

      EXX
      PUSH    HL
      LD      HL, #58B1
      LD      C, 14
K_1   LD      (L1+1), HL
      LD      DE, #0008
      ADD     HL, DE
      LD      (L2+1), HL
      LD      DE, #0018
      ADD     HL, DE
      LD      B, 8
K0    EXX
      LD      HL, L2-3
      LD      B, 5
K1    LD      A, (IX+0)
      INC     IX
      LD      (HL), A
      INC     HL
      LD      A, (IX+0)
      INC     IX
      LD      (HL), A
      DEC     HL
      DEC     HL
      DEC     HL
      DEC     HL
      DEC     HL
      DJNZ    K1
      LD      BC, L2-L1
      LD      HL, L1
      LDIR
      LD      B, 4
      LD      HL, L3-5
K2    LD      A, (IX+0)
      INC     IX
      LD      (HL), A
      INC     HL
      LD      A, (IX+0)
      INC     IX
      LD      (HL), A
      DEC     HL
      DEC     HL
      DEC     HL
      DEC     HL
      DEC     HL
      DJNZ    K2
      LD      BC, L3-L2
      LD      HL, L2
      LDIR
      EXX
      DJNZ    K0
      DEC     C
      JR      NZ, K_1
      EXX
      LD      B, L4-L3
      LD      HL, L3LDIR
      LD      HL, INT
      LD      (#94FF), HL
      LD      (FOR_SP), SP
      LD      A, #94
      LD      I, A
      IM      2
      EI
      HALT
      LD      BC, 1026

```

```

K3      DEC     BC
        LD      A,B
        OR      C
        JR      NZ,K3
        LD      BC,(0)          ; !!!
        NOP                    ; !!!
        NOP                    ; !!!
        CALL   L4
        POP    HL
        EXX
        IM     0
        EI
        RET
INT     LD      BC,(0)          ; !!!
        LD      BC,(0)          ; !!!
        RET
FOR_SP  DEFW    0
L1     LD      SP,#0000
        LD      DE,#0000
        PUSH   DE
        LD      DE,#0000
        PUSH   DE
L2     LD      SP,#0000
        LD      DE,#0000
        PUSH   DE
        LD      DE,#0000
        PUSH   DE
        LD      DE,#0000
        PUSH   DE
        LD      DE,#0000
        PUSH   DE
        ADD    HL,HL           ; !!!
        NOP                    ; !!!
L3     LD      B,1790
L31    DEC     BC
        LD      A,B
        OR      C
        JR      NZ,L31
        LD      R,A           ; !!!
        INC    DE             ; !!!
        LD      A,#7F
        IN     A,(#FE)
        RRCA
        JP     C,L4
        LD      SP,(FOR_SP)
        DEC    SP
        DEC    SP
        RET
DEFS   4954
END    NOP

```

Программа аналогична предыдущей, только тут применен иной принцип работы с экраном. Прямоугольник multicolor'a отцентрирован и занимает 16x16 знакомест. Атрибуты (2048 байт) должны располагаться с адреса SCR по 16 байт на строку.

```

;16X16 MULTICOLOR
        ORG    #8000
SCR     EQU    #C000
        ENT    $

```

```

    EXX
    PUSH    HL
    LD      HL, #5888
    LD      IX, L4
    LD      A, 16
K0    LD      C, 8
    LD      DE, #0004
K1    LD      B, 8
    PUSH    HL
K2    LD      (IX+0), #E1
    LD      (IX+1), #22
    LD      (IX+2), L
    LD      (IX+3), H
    ADD     IX, DE
    INC     HL
    INC     HL
    DJNZ    K2
    LD      B, E
K3    LD      (IX+0), D
    INC     IX
    DJNZ    K3
    POP     HL
    DEC     C
    JR      NZ, K1
    LD      BC, #0020
    ADD     HL, BC
    DEC     A
    JR      NZ, KO
    PUSH    IX
    POP     DE
    LD      BQL3-L1
    LD      HL, L1
    LDIR
    LD      HL, INT
    LD      (#92FF), HL
    LD      (FOR_SP), SP
    LD      A, #92
    LD      I, A
    IM      2
    EI
    HALT
K4    LD      BC, 956
    DEC     BC
    LD      A, B
    OR      C
    JR      NZ, K4
    LD      BC, (0)      ; !!!
    NOP                    ; !!!
    CALL    L3
    POP     HL
    EXX
    IM      1
    EI
    RET
INT
FOR_SP  DEFW    0
L1      LD      BC, 1652
L2      DEC     BC
    LD      A, B
    OR      C
    JR      NZ, L2
    LD      R, A      ; !!!
    LD      A, #7F
    IN     A, (#FE)

```

```
RRCA
JP      C, L3
LD      SP, (FOR_SP)
DEC     SP
DEC     SP
RET
L3      LD      SP, SCR
L4      DEFS   4633
END     NOP
```

P.S. Буквально на днях посмотрел "INSULT MEGADEMO" Code Busters. Полноэкранный скроллер в "секретной" части megademo не явился для меня откровением.

Ввиду всего вышеизложенного, теоретически вопрос ясен (хотя от теории до практики всегда далеко). Следует только заметить, что размеры scroller'a по горизонтали нельзя существенно улучшить, поскольку быстрая команда вывода на бордюр OUT (#FE),А длится 11 тактов, а реально применяемая OUT (C),reg8 -12 тактов, что представляет собой 12x24 точки экрана, то есть 3 знакоместа (что, похоже, и реализовано в megademo).

ПОЛУЧЕНИЕ СЛУЧАЙНЫХ ЧИСЕЛ

© А. Астафьев, г. Новосибирск

При решении многих прикладных задач возникает необходимость получения случайных чисел. С их помощью выполняются решения задач вычислительного анализа, когда получить точное решение трудно, и можно воспользоваться лишь методом приближенных вычислений (Метод Монте-Карло). Случайные числа используются при исследовании устойчивости программ в неожиданных ситуациях. Они, также, требуются во множестве моделирующих, обучающих, игровых и демонстрационных программ, не говоря уж о применении в сфере машинной графики - как видно спектр применения генераторов случайных чисел очень широк.

На первых порах вычислительной техники задачу получения случайных последовательностей чисел пытались решить, используя физические явления: электронный шум, радиоактивный распад и т.д. В иных случаях строили в памяти ЭВМ готовые таблицы случайных чисел. Но все это было неудачным. Таблицы быстро исчерпывались, а физические явления повторить было нельзя. Возникла парадоксальная ситуация - необходимо было получать случайные последовательности, но в то же время, чтобы можно было получать такие же последовательности еще раз!

Задача разрешилась после изобретения алгоритмов, в которых следующее случайное значение вычислялось из предыдущего. Такие алгоритмы обычно называют генераторами псевдослучайных чисел, а числа или последовательности, получаемые таким образом, назвали псевдослучайными, или проще - случайными.

В настоящее время известно множество алгоритмов получения псевдослучайных чисел (ПСЧ). Между собой они различаются способами формирования следующего ПСЧ из предыдущего - и отсюда разными свойствами ПСЧ - последовательностей. Основными параметрами генераторов (или как их еще называют - датчиков) ПСЧ являются следующие:

1. Период повторения последовательности.
2. Равномерность последовательности
3. Диапазон чисел (масштаб).
4. Разброс последовательности значений чисел.

Период повторения задаётся обычно как степень двойки, и, как правило, целесообразно выбирать его не больше, чем количество обращений к алгоритму за обозримый (требуемый) промежуток времени. На практике период 2 в 16-й вполне достаточен.

Под равномерностью понимается то, что вероятность возможного числа N-последовательности в ее начале, близка к вероятности появления N в ее конце. Говоря проще, мы можем иметь ПСЧ последовательность из 10000 чисел и в первых 9500 числах никогда не встретить Число 13. Если генерируются числа от 0 до 255, то это довольно плохо.

С масштабом все просто - масштаб можно изменить простейшими математическими преобразованиями.

Разброс значений имеет довольно большое значение - у хорошего генератора ПСЧ он должен быть как можно более хаотичен. Вообще, говоря правильными терминами, хороший генератор имеет равномерный спектр последовательности, и сам спектр желательно иметь похожим на спектр шумового сигнала.

Исследовать спектр получаемой с помощью генераторов ПСЧ - последовательности я предлагаю так: вызвать генератор ПСЧ, и на каждом вызове присваивать сначала X, а затем Y-координате точки значения генератора. Попробуйте такой эксперимент:

```
10 BORDER 0: INK 7: PAPER 0: CIS
20 RANDOMIZE 1
30 LET X=255*RND
40 LET Y=175*RND
50 PLOT X, Y: GO TO 30
```

Запустите все это и подождите минут пять. Заметили несовершенство этой программки? Точки норовят выстраиваться вовсе не случайно, а именно вертикальными столбцами. А теперь попробуйте ввести следующую строку:

```
35 LET Z=RND
```

и еще раз запустите программу. Заметили разницу? Да, из-за холостого вызова генератора в 35 строке эффект уже другой - нет столбиков из точек. Эти столбики возникают в результате особенности примененного в ПЗУ Спектрума генератора RND - у него хоть и более-менее равномерная последовательность, но вот разброс значений (пункт4) далеко не "шумовой". Это, кстати, стоило бы учитывать. Подобные особенности генераторов ПСЧ весьма критичны в некоторых сферах их применения - скажем, если генератор используется для имитации кубика в игре типа нарды, кости, домино и т.д., то ситуации выпадения четверки за тройкой, шестерки за пятеркой и т.д. весьма могут быть предсказуемыми из-за использования плохих датчиков ПСЧ.

Или представьте себе - процедура на ассемблере имитирует звездное небо. Генератор плохой, и вместо неба у вас получаются полоски...

Теперь рассмотрим алгоритм получения ПСЧ в родном спектрумовском ПЗУ. Алгоритм таков, судя по данным из литературы:

seed=seed+1

seed=seed+75

seed=seed mod 65536 [65537?]

Bask-переменным значение seed возвращается как seed/65537. Хотя подобные данные следовало бы проверить, я не вполне уверен в точности данного алгоритма изменения seed. Но это не является моей целью, и я оставляю возможность выяснить алгоритм работы ПЗУ-шного датчика ПСЧ читателям. Более интересным фактом является то, что ПЗУ-шный алгоритм работает по хорошо известному алгоритму линейного конгруэнтного генератора. Только в ПЗУ, судя по всему, он несколько изменен по сравнению с оригиналом. Так вот, формула генерации следующего числа для линейного конгруэнтного генератора такова:

$X(j+1) = (X(j)*A+B) \bmod R$, где

X(j) - текущий элемент ПСЧ-последовательности;

A и B - константы;

R - константа, задающая период;

a mod b - остаток от целочисленного деления a на b.

Этот генератор имеет максимальный период последовательности R до того, как она начнет повторяться. Период последовательности зависит от констант A и B. Линейный конгруэнтный генератор имеет максимальную длину последовательности ПСЧ тогда и только тогда, когда $A \bmod 4 = 1$ и B - нечетное. Это математически доказано.

Теперь попробуем реализовать такой датчик на ассемблере.

```
RND      LD      DE, 0
          LD      A, 77
          CALL   MUL
          INC    HL
          LD      (RND+1), HL
          RET
MUL      LD      HL, 0
MULTS   AND     A
          RET     Z
          RRA
          JR     NC, $+3
          ADD   HL, DE
          SLA   E
          RL   D
          JP   MULTS
```

Эта процедура возвращает в HL псевдослучайные числа согласно алгоритму линейного конгруэнтного датчика ПСЧ. Здесь константа A=77, а B=1, чтобы удовлетворить максимальному периоду последовательности, который, кстати, равен длине регистров, и составляет $2^{16}=65536$ двухбайтных чисел. Операция MOD выполняется автоматически при умножении, так как мы теряем (усекаем) результат, оставляя лишь младшие его разряды, т.е. и выполняем функцию mod. Ради экспериментов можно менять число 77 на какое-либо свое. Важно лишь помнить, что остаток от деления этого числа на 4 (целочисленного деления) должен быть равен 1. Вместо постоянного коэффициента B=1 - это просто INC HL; можно конечно устроить:

```
LD      DE, X
ADD     HL, DE
```

где X следует брать нечетным.

Таким образом, изменются константы, и следя, чтобы они удовлетворяли требованию максимального периода последовательности, можно получать разные псевдослучайные последовательности. Если необходимо получить байт ПСЧ, то лучше всего брать его из старших разрядов - в данном случае из регистра H, так как период повторения кода в старших разрядах больше чем в младших. По этой же причине рекомендую, если необходимо менее 8 разрядов, пользоваться сдвигом вправо, а не усечением командой AND. Замечу, что процедуру умножения можно сделать на 2 байта короче, изменив последний JP на JR и SLA E, RL D на:

```
EX      DE, HL
ADD     HL, HL
EX      DE, HL
```

но потеряв при этом на каждом цикле умножения целых 5 тактов Z80. Кстати, циклов умножения необходимо 8, и можно вместо проверки множителя на 0 и досрочного выхода из процедуры просто поставить цикл. А можно, переписав процедуру, получить следующее:

```
MUL      LD      HL, 0
```

```

MULT   RR   A
        JR   NC, $+3
        ADD  HL, DE
        EX   DE, HL
        ADD  HL, HL
        EX   DE, HL
        JR   NZ, MULT
        RET

```

Однако, вернемся к нашему генератору ПСЧ. Если положить $A=5$ и $B=1$, что удовлетворяет условиям получения максимально длинной последовательности, то можно написать линейный конгруэнтный генератор и проще, пожертвовав при этом лучшей равномерностью последовательности:

```

RND    LD    HL, 0
        LD    D, H
        LD    E, L
        ADD  HL, HL           ; HL=HL+2
        ADD  HL, HL           ; HL=HL+4
        ADD  HL, DE           ; HL=HL+5
        INC  HL               ; HL=HL+1
        LD    (RND+1), HL
        RET                    ; в HL будет новое ПСЧ, в DE - старое ПСЧ.

```

Теперь я хочу предложить несколько датчиков ПСЧ. Они хотя и не имеют математически обоснованных алгоритмов своей работы, но работают, тем не менее, неплохо:

```

RND    LD    HL, 0
        LD    A, H
        ADD  A, #77
        LD    H, A
        RLC  L
        ADD  A, L
        LD    L, A
        LD    (RND+1), HL
        RET                    ; в A - готовое ПСЧ.

```

Можете поэкспериментировать с этим генератором - попробуйте изменять число в 3-й строке, или команду `RCL L` на `RRC L`; `ADD` на `SUB` и так далее. Кстати, в отличие от предыдущих процедур, здесь нет такого правила, что период повторения кода в старших разрядах больше чем в младших, поэтому запросто можно пользоваться для усечения результата командой `AND XX`. Период последовательности у этой процедуры около 64000.

Следующая процедура оставляет в трех своих ячейках сразу три ПСЧ:

```

RND    LD    HL, SEED
        LD    A, (HL)
        RRCA
        RRCA
        RRCA
        AND  #1F
        INC  HL
        XOR  (HL)
        INC  HL
        RRA
        RL  (HL)
        DEC  HL
        RL  (HL)
        DEC  HL
        RL  (HL)           ; в A - ПСЧ, HL указывает на SEED.
        RET
SEED   DEFB  128, 255, 63

```

Теперь приведу процедуру, интересную тем, что у нее можно изменять равномерность последовательности:

```

RND    LD    HL, 0
        LD    B, 23           ; константа глубины.
INRND LD    A, H
        ADD  HL, HL
        AND  A
        JP   PO, $+4

```

```

INC     HL
DJNZ   INRND
LD     (RND+1),HL ; HL содержит ПСЧ.
RET    ; A тоже содержит ПСЧ.

```

Однако, из-за того, что процедура имеет внутренний цикл, работает она сравнительно медленно.

И, наконец, приведу процедуру, которую, в отличие от всех предыдущих процедур написал не я сам, а DAVE PERRY, автор известной на SEGA игры Alladin, и знакомый нам на Спектруме играми Savage, Turtles и т.д. Я нашел эту процедуру в играх Savage, Stainles Steel. Выглядит она так:

```

RND     PUSH     HL
        PUSH     HL
        LD      DE, (SEED)
        LD      H, E
        LD      L, #FD
        LD      A, D
        AND     A
        SBC    HL, DE
        SBC    A, 0
        SBC    HL, DE
        SBC    A, 0
        LD      E, A
        LD      D, 0
        SBC    HL, DE
        JR     NC, $+3
        INC    HL
        LD      (SEED), HL
        POP    DE
        POP    HL ; в A и B SEED останется ПСЧ.
        RET
SEED    DEFW    0

```

Стоит, кстати, заметить, что если требуется ПСЧ в районе 0...127, то его с успехом можно считать из регистра R процессора. Правда, такой трюк будет проходить только, если между считываниями ПСЧ из регистра R, основная программа выполняет процедуры с изменяющимися временными задержками или если пользователь внешне заставит программу выполнять какие-либо действия. Если же попробовать использовать регистр R для получения ПСЧ в каком-либо коротком цикле, то особенно надеяться на успех не приходится. Также, можно попробовать использовать, как таблицу готовых ПСЧ, само ПЗУ Спектрума. Этот метод, в сочетании с регистром R, уже эффективнее, и им пользуются. Это можно сделать, например, так:

```

RND     LD      HL, 0 ; текущий адрес ПЗУ
        INC    HL
        LD      A, H
        AND    #1F ; не #3F, т.к. в конце ПЗУ находится фонт.
        LD      H, A
        LD      (RND+1), HL
        LD      A, R
        XOR    (HL) ; для большей случайности.
        RET

```

Но гораздо эффективней пользоваться другими генераторами ПСЧ - например, линейным конгруэнтным генератором и т.п. Если Вам потребуется небольшое знаковое ПСЧ, то его можно получить, например, так:

```

CALL   RND ; в A имеем ПСЧ
AND    7 ; сделаем его не более 7
SUB    4
RET    ; теперь ПСЧ в ранге -4...+3

```

В общем и целом, данная статья дает возможность использовать эффективные генераторы ПСЧ в самых различных областях программирования. Можно, скажем, защищать (XOR-ить) свои файлы, как я сделал в импортированной мною польской мегадемо LSD (напомню, что после таких XOR-ок файлы будут очень плохо сжиматься) или использовать генераторы ПСЧ в процедурах вывода звука (шума), и т.д. Напоследок приведу три примера процедур, использующих генератор ПСЧ:

Пример 1.

```

ORG     #8000
STARS  EI
        HALT
        CALL   RND
        LD     C, A

```

```

RE_RND  CALL  RND
        CP    192
        JR    NC, RE_RND
        LD    B, A
        CALL  #22B1
        INC  A
        LD    B, A
        LD    ъA, 1
        RRCA
        DJNZ  $-1
        OR   (HL)
        LD   (HL), A
        LD   A, #7F
        IN  A, (#FE)
        RRA
        JR   C, STARS
        RET
RND     LD    HL, 0
        LD    A, H
        ADD  A, #77
        LD    H, A
        RLC  L
        ADD  A, L
        LD    L, A
        LD   (RND+1), HL
        RET

```

Пример 2.

```

                ORG    #8000
NOISEFX DI
        SUB    A
INNER_L LD    D, H
        LD    E, L
        ADD  HL, HL
        ADD  HL, HL
        ADD  HL, DE
        INC  HL
        JR   C, INNER_L
        XOR  #1B
        OUT  (#FE), A
        JR   INNER_L
ZX_HANG EI
        HALT
        LD    DE, #5800
INNER  CALL  RND
        SRL  A
        LD   (DE), A
        INC  DE
        LD   A, D
        CP   #5B
        JR   C, INNER
        JR   ZX_HANG
RND    LD    HL, 0
        LD    A, H
        ADD  A, #77
        LD    H, A
        RLC  L
        ADD  A, L
        LD    L, A
        LD   (RND+1), HL
        RET

```

Пример 3.

```

                ORG    #8000
B_FX    EI
        HALT

```

```

C_OUT  LD      HL, (RND+1)
        PUSH   HL
        LD      DE, 210
        CALL   RND
        AND     7
        OUT    (#FE), A
        LD      B, 14
        DJNZ   $
        DEC    DE
        LD      A, D
        OR     E
        JR     NZ, C_OUT
        POP    HL
        LD      (RND+1), HL
        CALL   RND
        LD      A, #7F
        IN     A, (#FE)
        RRA
        JR     C, B_FX
        RET

RND     LD      HL, 0
        LD      A, H
        ADD    A, #77
        LD      H, A
        RLC   L
        ADD    A, L
        LD      L, A
        LD      (RND+1), HL
        RET

```

Процедура NOISE_FX использует внутренний генератор ПСЧ, а остальные - процедуры генераторов из первой. Вы можете по своему вкусу установить цвета, режим прерываний, опрос клавиш и т.д.

КОМПЬТЕРЫ, КОТОРЫЕ МЫ ВЫБИРАЕМ



© Шилиев Павел (Stellar company inc), г. Кирово-Чепецк, 1996.

ДОСТУП К ВЕРХНЕЙ ОБЛАСТИ ПАМЯТИ.

Я пишу вам по поводу решения проблемы с различными способами доступа к верхней памяти в различных машинах. Ведь не секрет, что сейчас появляется много программ, использующих верхнюю память, но они ориентированы, в основном, на один тип компьютера. Но как же быть пользователям других компьютеров?

Первое решение этой проблемы я вижу в создании специального файла RAM_DRIVE. Пусть в нем хранится подпрограмма, обслуживающая верхнюю память компьютера. Все, что нужно создателю программы, так это объяснить какие параметры должна эта подпрограмма использовать, где располагаться и т. д., тогда каждый пользователь, умеющий программировать, напишет свой вариант подпрограммы RAM_DRIVE.

Другой способ я вижу такой: пусть все пользователи компьютеров с расширенной памятью дадут исчерпывающую информацию о своем компьютере на страницах "ZX-РЕВЮ". Тогда создатель ПО по этим данным сможет адаптировать свое творение ко многим машинам.

В свою очередь, я предлагаю информацию о компьютере PROF1.

Памятью он оперирует с помощью двух портов: CMR1 (#DFFD) и CMR0 (#7FFD). Порт CMR0 ничем особым не отличается от порта в обычном 128 компьютере, поэтому рассмотрим "поближе" порт CMR1 и назначения его битов:

D0 \uparrow

D1 \blacktriangleright Номер проецируемой страницы памяти.

D2 \downarrow

D3 \blacktriangleright Выбор положения окна проецирования сегмента памяти:

0 - окно N1 (#C000 - #FFFF)

1 - окно N2 (#4000 - #7FFF)

D4 \blacktriangleright блокировка ПЗУ, на его место включается страница 0.

D5 \blacktriangleright 1 - блокировка контроллера ВГ93, включение его портов на доступ из ОЗУ.

D6 \blacktriangleright проецирование дополнительного экрана SEG 06 в карту памяти процессора на место SEG 02.

D7 \blacktriangleright 0 - экран SPECTRUMa

1 - экран PROF1.

Как же происходит переключение? Вы выдаете в порт #7FFD номер сегмента (как на обычном 128) - это будет столбец (см. таблицу), а в порт #DFFD выдаете строку (разряды D0-D2). И по этим данным выбирается страница. Это можно проиллюстрировать следующей таблицей:

		CMR0							
		0	1	2	3	4	5	6	7
C M R 1	0	00	01	02	03	04	05	06	07
	1	08	09	10	11	12	13	14	15
	2	16	17	18	19	20	21	22	23
	3	24	25	26	27	28	29	30	31
	4	32	33	34	35	36	37	38	39
	5	40	41	42	43	44	45	46	47
	6	48	49	50	51	52	53	54	55
	7	56	57	58	59	60	61	62	63

Здесь условно показана память компьютера PROF1, т.е. 00,01,...,63 - это номера банков памяти. Пример: выдаем в порт CRM0 байт 0xxxxx000, а в порт CMR1 байт 06, тогда с адреса #C000 у нас будет стоять банк 48.

В таблице показана память PROF1 1024К, но как эта память выглядит на PROF1 512К и как с ней оперировать? На самом деле все очень просто! В компьютерах с памятью меньше, чем 1024К, некоторые "банки" памяти отсутствуют (в моем компьютере, например, нет "банков" памяти с 16 по 47), а как обращаться к остальным "банкам" памяти мы рассмотрели выше. Только учтите, если выбрать отсутствующий "банк" памяти, то с адреса #C000 будут числа #FF, и записать туда ничего нельзя. Этим фактом можно пользоваться при составлении карты памяти компьютера.

Внимание!!! Если Ваша программа рассчитана на PROF1, то перед началом работы проверьте наличную память, или предложите пользователю самому составить таблицу распределения "банков" памяти. Это необходимо, так как отсутствовать могут любые "банки" памяти.

Теперь насчет того, как определить наличие порта CMR1. Для этого можно заблокировать ПЗУ и записать туда что-нибудь. Если запись прошла успешно, то порт CMR1 имеется в наличии. Другой способ: заносим по адресу #C000 число 1 и подаем в порт #DFFD байт 01. Если после этого из адреса #C000 считано другое число, то в Вашем компьютере также есть порт CMR1.

ЭТЮДЫ

© А.Астафьев, г.Новосибирск

КОРР: То, что вы назвали "гасилкой экрана", называется FADE-OUT эффектом (это принято во всем мире), а "зажигалка", соответственно, называется FADE-IN эффектом.

Проанализировав предыдущие процедуры, я пришел к выводу, что все забыли об одном замечательном методе программирования - табличном программировании. А зря. Порой, метод табличного программирования дает изящное, простое и, главное, быстродействующее решение. Ниже, как иллюстрацию данного метода, я привожу FADE-OUT, гасящую экран "как в TERMINATOR'e".

```
ORG      #8000
PREPARE LD      HL,DAL_TBL
PREP_T   LD      A,L
        AND     7
        JR      Z,$+3
        DEC    A
        LD     E,A
        LD     A,L
        AND   #38
        JR     Z,$+4
        SUB   8
        OR    E
        LD    (HL),A
        INC  L
        JR   NZ,PREP_T
FADEOUT LD      D,DAL_TBL/256 ; в D - старший байт адреса таблицы
        LD     B,7           ; ровно 7 переходов
W_RAY   HALT                    ; ждем сканирования экрана
        LD     HL,#5800      ; это адрес атрибутов
ATR_DEC LD      E,(HL)        ; берем код атрибута
        LD     A,(DE)
        LD     (HL),A        ; и заменяем его значением из таблицы
        INC   HL             ; повторяем 768 раз до конца области атрибутов
        LD     A,H
        CP    #5B
        JR    C,ATR_DEC
        DJNZ  W_RAY          ; повторяем 7 раз на случай белого цвета
        RET
ORG     $&#FF00+256 ; выравниваем адрес таблицы на 256
DAL_TBL DEFS   256 ; резервируем 256 байт
```

Пару слов по данной процедуре. Написана она в TASM128, который легко дает возможность определить адреса таблиц, кодов и т.п. более простыми методами, чем ZEUS. Сама же процедура создает для своей работы таблицу из 256 байт. Таблица располагается по адресу, выровненному на 256 - с такими таблицами процессору проще работать, так как от элемента к элементу в таблице можно переходить командами INC L или DEC L (предполагаем, что адрес таблицы находится в HL), а от таблицы к таблице переходить командами INC H / DEC H. Аналогичным образом, элементарно находятся и элементы в таких таблицах. Порой, достаточно загрузить номером элемента младшую часть адресного регистра. Вдобавок ко всему, подобные таблицы (если, конечно, занимают все 256 байт таблицы) обладают способностью "заворота". Вот простой пример:

```
POINTER LD      HL,#BE00      ; смещение на элемент таблицы
        LD     A,STEP         ; где STEP знаковое!
        ADD   A,L
        LD     L,A
        LD     (POINTER+1),HL
        LD     A,(HL)
        ...
```

Здесь каким бы ни было значение STEP - мы можем "летать" по таблице, перескакивая хоть по 50 элементов, но никогда не вырвемся за ее пределы!

Если требовался бы двухбайтный результат - легко продолжить:

```
INC     H
LD      H,(HL)
LD      L,A
```

...

Но вернемся-таки к нашей процедуре FADE-OUT. Если не быть буквоедом, пожертвовать памятью, то можно написать процедуру - рекордсмен по длине. Пусть дополнительно использованы 2048 байтов, пусть предварительно нужно прогнать процедуру, которая выстраивает эту таблицу, но зато какой результат! И все как в TERMINATOR'e!

```
ORG      #8000
PAL_TBL EQU    0-256*8
FADEOUT LD     D,PAL_TBL/256
W_RAY   HALT
LD      H, #5A
ATR_DEC LD     E, (HL)
LD      A, (DE)
LD      (HL), A
DEC     HL
BIT     3, H
JR      NZ, ATR_DEC
INC     D
JR      NZ, W_RAY
RET
PREPARE LD     HL, PAL_TBL
LD      BC, #0800
FILL_T  LD     A, L
AND     7
JR      Z, $+3
DEC     A
LD      E, A
LD      A, L
AND     #38
JR      Z, $+4
SUB     8
OR      E
LD      (HL), A
CPI
JP      PE, FILL_T
RET
```

Сама процедура FADE-OUT занимает 17 байт. Для приготовления таблицы можно использовать несколько модифицированную PREPARE. И под конец, для затравки умов читателей, приведу пример процедуры установки точки на экран, которая, во-первых, сама по себе работает очень быстро, и, во-вторых, также является иллюстрацией табличных методов программирования. Из особенностей этой процедуры приведу следующую - не требуется никаких дополнительных вычислений, чтобы ограничить размер экрана по Y-координате. Вследствие того, что для точек с координатами больше 191, таблица просто заполнена нулями - точка просто не ставится на экран, процедура посылает данные в адрес 0...31, где находится ПЗУ. Таблица занимает 1К памяти и, конечно же, выровнена на адрес, кратный 256. Я называю такие адреса - "адресами начала параграфа", где параграф - блок 256 байтов с адреса, кратного 256. Вообще, во всех 8-битных процессорах такие блоки называют страницами (64К=256 страниц по 256 байт), но я, чтобы избежать путаницы со страницами 128-ого Спектрума, называю их параграфами.

```
PSET    LD     H, SCR_TBL/256 ; H - начало таблицы
LD      L, E ; в L загружаем X-координату
LD      C, (HL) ; берем байт с необходимым битом
INC     H ; шагаем к следующему параграфу
LD      A, (HL) ; берем значение X/8 - это номер столбца
INC     H
LD      L, D ; теперь нужна Y-координата
OR      (HL) ; HL указывает сейчас на младшую часть адреса точки
INC     H
LD      H, (HL) ; берем старшую часть адреса
LD      L, A ; в HL - адрес точки, в C - нужный бит
LD      A, (HL) ; ставим точку по OR
OR      C
LD      (HL), A
RET
SET_TBL LD     HL, SCR_TBL ; в HL - адрес таблицы
```

```

LD      B,L          ; в B=00, так как таблица выровнена
LD      A,#80        ; начинаем от старшего бита
SET_DOT LD      (HL),A ; и заносим его в список
RRCA    ; переходим к следующему биту
INC     HL
DJNZ   SET_DOT      ; и так 256 раз
SET_DIV LD      A,L
RRCA    ; три вращения для деления на 8
RRCA
RRCA
AND     31          ; а можно бы было дать три раза SRL A
LD      (HL),A      ; так мы заносим в (HL) значение L деленное на 8
INC     HL
DJNZ   SET_DIV      ; и так 256 раз
LD      B,#C0        ; требуется 192 экранных адреса
LD      DE,#4000     ; адрес левого столбца - самая верхушка
SET_ADR LD      (HL),E
INC     H
LD      (HL),D       ; грузим DE в два параграфа. E в первый; D - во
                          ; второй
DEC     H
INC     L
CALL   D_STEP        ; смещаем на байт (точку) ниже
DJNZ   SET_ADR       ; повторим 192 раза
RET
D_STEP INC     D       ; смещаемся на 256 в экране - байт вниз
LD      A,D          ; проверка - это следующее знакоместо?
AND     7
RET     NZ           ; если нет - вернемся обратно
LD      A,E          ; сместимся на знакоместо вниз (+32)
ADD     A,32
LD      E,A
RET     C            ; должны перейти в следующую треть - возвращаемся
LD      A,D          ; иначе компенсируем результат восьми INC D
SUB     8            ; возвращаем адрес в текущую треть экрана
LD      D,A          ; (треть, кстати, равна 256*8=2048 (256=INC D))
RET
ORG     $&#FF00+256
SCR_TBL DEFS    256*4 ; резервируем 1K нулей средствами TASM128

```

Здесь процедура SET_TBL подготавливает эту таблицу. Сама таблица выглядит следующим образом:

- #8100 256 байт - байты с установленным битом в соответствии с X.
- #8200 256 байт - значение в таблице = ее индекс/b (b - номер столбца (X)).
- #8300 192 байта в начале - младший байт адреса левого столбца экрана.
- #8400 первые 192 байта - аналогично, старшие байты.

Вот такие вот "пирог". Думаю, недолго домыслить, что с начала параграфа удобно хранить не только таблицы. Там можно размещать спрайты, фонт (что те же самые спрайты и есть), многобайтные переменные (для этого группируйте их), и т.д. и т.п. Кстати, не забывайте, что родной фонт размещается с адреса #3D00 – суть начала параграфа. Аналогично, и экран Спектрума тоже.

© И. Пивен.

КОРР: Решил вынести на ваш суд свою первую (законченную) процедуру в машинных кодах. Сей релоцируемый программный модуль позволяет выводить на экран 4 любых символа в восьмикратном увеличении (8*4=32, как известно). Надеюсь, на основе моей идеи читатели создадут "BIG RUNLINE" - это, вообще-то, цель письма. Итак, вперед:

```

; (C) SELINA SPACE 8*8
ORG     23296        ; можно и любой другой адрес
EXX    ; подготовка для файла атрибутов
LD      HL,22520
LD      BC,8
EXX
CALL   124          ; RET - обеспечение релацируемости и

```

```

DEC     SP           ; расчет адреса сообщения
DEC     SP
POP     HL
LD      BC,108
ADD     HL,BC       ; получаем STRING
FIR_NEX LD      A,(HL) ; подготовка или возврат, если END
CP      255
JR      NZ,CONT
LD      HL,10072
EXX
RET
CONT    INC     HL           ; расчет адреса матрицы символа
PUSH   HL
LD      L,A
LD      H,0
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
PUSH   HL
POP     BC
LD      HL,(23606)
ADD     HL,BC
PUSH   HL
POP     DE
EXX
ADD     HL,BC         ; расчет адреса в файле атрибутов
PUSH   HL
LD      B,8          ; количество "огромных" знаколиний для цикла
NEX_LIN PUSH  BC
LD      A,(DE)       ; байт матрицы в аккумулятор и проверяя каждый бит,
INC     DE           ; включаем или не включаем соответствующий байт
BIT    7,A           ; атрибутов.
JR      Z,AA
LD      (HL),214
AA      INC     HL
BIT    6,A
JR      Z,BB
LD      (HL),214
BB      INC     HL
BIT    5,A
JR      Z,CC
LD      (HL),214    ; 214 - параметры (атрибуты) знакоместа
CC      INC     HL
BIT    4,A
JR      Z,DD
LD      (HL),214
DD      INC     HL
BIT    3,A
JR      Z,EE
LD      (HL),214
EE      INC     HL
BIT    2,A
JR      Z,FF
LD      (HL),214
FF      INC     HL
BIT    1,A
JR      Z,GG
LD      (HL),214
GG      INC     HL
BIT    0,A
JR      Z,HH
LD      (HL),214
HH      LD      BC,25       ; переход к следующей "огромной" знаколиннии
ADD     HL,BC

```

```

POP      BC          ; восстанавливаем параметр цикла
DJNZ    NEX_LIN     ; не закончили? Тогда на NEX-LINE
POP      HL          ; адрес следующего символа вернули на FIR_NEX
JR      FIR_NEX     ; начнем сначала.

```

КОРР: Конечно, не забываю рубрику "ИДЕЯ" и дарю довольно интересную мысль читателям.

Вероятно, с самых первых шагов общения с RAM, каждый сталкивается с областью системных переменных, и, в первую очередь, с CHARS. Ну, конечно, у Вас уже мелькнула стереотипная мысль: "CHARS указывает на 256 байтов ниже истинного положения фонта". А почему?! Любой Ваш ответ неверен! На самом деле, CHARS указывает туда, куда надо! Просто процедура печати из ПЗУ не изображает символы с кодами 0...31. Ага! Начинает доходить? Ведь это нужно для работы БЕЙСИКА! А в кодах? Вы собственноручно создаете процедуру печати, которая не игнорирует "контрольные коды" и, таким образом, получаете вдобавок еще 32 символа в Вашем фонте. Создать символы можно в ART STUDIO, указать в CHARS непосредственно на них, а остальной набор загрузить на 256 байтов выше. И все! На базе этой идеи можно создать или доработать текстовый редактор, который, думаю, заинтересовал бы многих.

© Васильев Антон (EARTH SOFTWARE) г. Снежинск

КОРР: Здравствуй уважаемый ИНФОРКОМ! Посылаю вам пару программ - ответов на вопросы Сергея Колотова, а также несколько своих программ.

Итак, сначала ответы:

1. HL= B*C

```

LD      E, C
LD      HL, 0
LD      D, 0
L1     ADD  HL, DE
DJNZ    L1
RET

```

2. Напечатать символ, код которого в А, атрибутом, заданным в С, на весь экран.

```

ORG     40000
SUB     32
LD      L, A
LD      H, 0
LD      A, C
LD      (L6+1), A
ADD     HL, HL
ADD     HL, HL
ADD     HL, HL
LD      DE, 15616
ADD     HL, DE
LD      DE, BUF
LD      BC, 8
LDIR
LD      HL, 22528
LD      DE, BUF
LD      B, 8
L1     PUSH BC
      PUSH DE
      PUSH HL
LD      B, 8
L2     PUSH BC
      EX  DE, HL
      RL (HL)
      EX  DE, HL
LD      A, 0
JR      NC, L3
L6     LD  A, 34
L3     LD  B, 4
L4     LD  (HL), A
      INC HL
      DJNZ L4
      POP BC
      DJNZ L2
      EX  DE, HL

```

```

        POP     HL
        PUSH    HL
L5      LD      B, 64
        LD      A, (HL)
        LD      (DE), A
        INC     HL
        INC     DE
        DJNZ   L5
        POP     HL
        LD      DE, 96
        ADD    HL, DE
        POP     DE
        INC     DE
        POP     BC
        DJNZ   L1
        RET
BUF     DEFS   8

```

Ну, а теперь я предлагаю свои программы: Первая - это индикатор, показывающий в виде полосок с периодически изменяющимся сдвигом на экране, текущую амплитуду звука в каналах АУ - СНР. Похожая разработка уже публиковалась на страницах ZX РЕВЮ, но в ней цвета были жестко заданы.

```

; ES FOR INFORCOM
        ORG    38000
        LD     HL, 22528
        LD     A, 8
        LD     B, 3
L1      LD     DE, BUF
        PUSH   DE
        PUSH   AF
        PUSH   HL
        PUSH   BC
        CALL  ANL
        LD     A, (PAUS)
        DEC   A
        JR    NZ, L2
        LD     B, 3
L4      LD     HL, BUF
        LD     A, (HL)
        INC   A
        CP    72
        JR    NZ, L3
L3      LD     (HL), A
        DJNZ  L4
        LD     A, 16
L2      LD     (PAUS), A
        POP   BC
        POP   HL
        LD     DE, 32
        ADD  HL, DE
        POP   AF
        INC   A
        POP   DE
        DJNZ  L1
        RET
ANL     LD     BC, #FFFD
        OUT   (C), A
        IN   A, (C)
        AND  15
L3      LD     B, 15
P3      AND   A
        JR    NZ, P1
        LD   (HL), 0
        JR   P2

```

```

P1      PUSH    AF
        LD      A, (DE)
        LD      (HL), A
        POP    AF
        DEC    A
P2      INC     HL
        DJNZ   P3
        RET
BUF     DEFB    65, 66, 67
PAUS    DEFB    16

```

Лучше всего, по моему, программа демонстрируется с мелодией из пятой части "SATISFACTION MEGADEMO" (там большую часть экрана занимает сетка). Мелодия располагается в нулевом банке (странице) ОЗУ с адреса 49152. Демонстрационная программа:

```

1 RANDOMIZE USR 15619: REM : LOAD "SAT"CODE
*** Загрузили мелодию ***
2 BORDER 0: PAPER 0: INK 7: CLS
*** Очистили экран и переинициализировали музыку на самое начало ***
3 REM
*** Рисуем сегменты индикатора ***
4 FOR N=0 TO 2
5 FOR K=0 TO 14: PRINT AT N,K; "-": NEXT K
6 NEXT N
7 REM
*** Проигрываем мелодию ***
8 RANDOMIZE USR 49158
9 RANDOMIZE USR 38E3
10 PAUSE 1
11 GO TO 8

```

Ну, уж коли, я завел разговор про "SATISFACTION MEGADEMO", то приведу еще одну свою разработку, на этот раз из серии "Бегущая строка". Приведенная ниже программа скроллирует по экрану спрайты размером 2*2 знакоместа, с возможностью окрашивания каждой литеры в свой цвет. Эффект похож на тот, который есть в той же части вышеупомянутого "MEGADEMO". В "листинге 1" я приведу саму б/с, а в "листинге 2" - программу формирующую для нее спрайтовый набор. Нарисуйте на экране буквы размером 2*2 знакоместа, как это указано в ZX-РЕВЮ 95/1 и запустите мою программу. Через секунду спрайтовый набор будет готов.

Листинг 1.

```

; ES FOR INFORCOM
        ORG    40000
        CALL  CH
        CALL  SCR
        CALL  ATR
        RET
CH      LD     A, (COL)
        AND   A
        JR    Z, CH1
        DEC  A
        LD   (COL), A
        RET
CH1     LD     A, 1
        LD   (COL), A
        LD   HL, (TEXP)
CH4     LD     A, (HL)
        CP   255
        JR   NZ, CH2
        LD   HL, 45000
        LD   A, (HL)
CH2     CP     8
        JR   NC, CH3
        ADD  A, 64
        LD   (CLR), A
        INC  HL
        JR   CH4
CH3     INC   HL

```

```

LD      (TEXP),HL
SUB     32
LD      H,0
LD      L,A
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
ADD     HL,HL
LD      DE,41000
ADD     HL,DE
LD      DE,BUF
LD      BC,32
LDIR
RET
SCR     LD      HL,16385
        LD      DE,16384
        LD      B,2
SC2     PUSH    BC
        PUSH    HL
        LD      B,8
SC1     PUSH    BC
        PUSH    HL
        PUSH    DE
        LD      BC,31
        LDIR
        POP     DE
        POP     HL
        INC     H
        INC     D
        POP     BC
        DJNZ   SC1
        POP     HL
        LD      DE,32
        ADD     HL,DE
        LD      D,H
        LD      E,L
        DEC     E
        POP     BC
        DJNZ   SC2
        LD      IX,BUF
        LD      HL,16415
        LD      B,2
SC4     PUSH    BC
        PUSH    HL
        LD      B,8
SC3     LD      A,(IX+0)
        LD      (HL),A
        LD      A,(IX+1)
        LD      (IX+0),A
        INC     IX
        INC     IX
        INC     H
        DJNZ   SC3
        POP     HL
        LD      DE,32
        ADD     HL,DE
        POP     BC
        DJNZ   SC4
        RET
ATR     LD      HL,22529
        LD      DE,22528
        LD      B,2
ATR1    PUSH    BC

```

```

        PUSH    HL
        LD      BC, 31
        LDIR
        POP     HL
        LD      DE, 32
        ADD    HL, DE
        LD      D, H
        LD      E, L
        DEC    E
        POP     BC
        DJNZ   ATR1
        LD      HL, 22559
        LD      A, (CLR)
        LD      B, 2
ATR2    LD      (HL), A
        LD      DE, 32
        ADD    HL, DE
        DJNZ   ATR2
        RET
COL     DEFB   0
TEXP   DEFW   45000
CLR     DEFB   7
BUF     DEFS   32
        ORG    45000
        DEFB   2
        DEFM   "HELLO EVERY BODY!"
        DEFB   6
        DEFM   "EARTH          SOFTWARE"
        DEFB   4
        DEFM   "PRESENTS ... "

```

...

Текст должен располагаться с адреса 45000. Он будет выводиться на экран с атрибутом повышенной яркости. Если это Вам не надо, то можно убрать строку ADD A,64. Спрайтовый набор располагается с адреса 41000 и формируется следующей программой.

Листинг 2.

```

        ORG    39000
; ES FOR INFORCOM
        LD      HL, 16384
        LD      DE, 41000
        LD      B, 4
        CALL   L4
        LD      HL, 18432
        LD      B, 2
        CALL   L4
        RET
L4     PUSH    BC
        PUSH    HL
        LD      B, 16
L3     PUSH    HL
        PUSH    BC
        LD      B, 2
L2     PUSH    HL
        PUSH    BC
        LD      B, 8
L1     PUSH    HL
        LD      A, (HL)
        LD      (DE), A
        INC    HL
        INC    DE
        LD      A, (HL)
        LD      (DE), A
        POP     HL
        INC    DE

```

```

INC      H
DJNZ    L1
POP     HL
POP     BC
PUSH    DE
LD      DE, 32
ADD     HL, DE
POP     DE
DJNZ    L2
POP     BC
POP     HL
INC     HL
INC     HL
DJNZ    L3
POP     HL
POP     BC
PUSH    DE
LD      DE, 64
ADD     HL, DE
POP     DE
DJNZ    L4
RET

```

Наконец, последняя программа - вывод на экран картинки следующим образом: картинка плавно и по-пиксельно, на протяжении всей высоты экрана, медленно раскрывается. В общем, как будто раскрывается одиночная горизонтальная штора. Если у картинки все атрибуты "PAPER" черные, то можно сперва вывести атрибуты, а уж потом монохромную часть.

```

; ES FOR INFORCOM
ORG     39000
LD      HL, ADR
LD      DE, 16384
LD      B, 32
L3     PUSH    HL
        PUSH    DE
        PUSH    BC
        XOR     A
        LD      C, 128
        LD      B, 8
L2     PUSH    HL
        PUSH    DE
        PUSH    BC
        RRCA
        OR      C
        LD      C, A
        LD      B, 192
L1     PUSH    BC
        AND     (HL)
        LD      (DE), A
        LD      B, 32
IC     INC     HL
        INC     DE
        DJNZ    IC
        POP     BC
        LD      A, C
        DJNZ    L1
        HALT
        POP     BC
        POP     DE
        POP     HL
        DJNZ    L2
        POP     BC
        POP     DE
        POP     HL
        INC     HL

```

```

INC     DE
DJNZ   L3
LD     HL,ADR+6911
LD     DE,22528
LD     BC,768
LDIR
RET

```

ADR

На страницах ZX РЕВИЮ достаточно подробно освещен музыкальный процессор AY 3-8910, но мне не ясно, что за параметр задается в регистрах 11-12 и как им пользоваться. И второй вопрос - как переделывать и воспроизводить музыку, голос и т.д. на Спектруме? Я знаю только то, что при воспроизведении применяются команды блочного вывода. Ну, а программы с использованием оцифровок, я думаю, называть не надо. Это SWIV, DIZZY-3, SATISFACTION MEGADEMO (там, кстати, у меня музыка не играет, т.к. RST7 не потрудился написать команды типа : LD BC,#7FFD: OUT (C),A. Хорошо хотя бы, что после этого программа вообще не глохнет!), LIRA II, MEGADEMO, LAMBADA, SABRINA...

© Кравников Денис, Украина.

КОРР: Странно, что в разделе "ЭТЮДЫ" я не встречал ни одной программы для создания различных эффектов на AY-3-8910. Привожу две программы:

1. Laser.

```

1 REM Laser; Regal Phoenix 1995
10 FOR I=0 TO 13: READ A: OUT 65533,I: OUT 49149,A: NEXT I
20 DATA 190,0,0,0,0,0,0,126,16,0,0,0,10,15

```

2. Выстрел.

```

1 REM SHOOTING; Regal Phoenix 1995
10 OUT 65533,6: OUT 49149,31
20 OUT 65533,7: OUT 49149,7
30 OUT 65533,8: OUT 49149,16
40 OUT 65533,9: OUT 49149,16
50 OUT 65533,10: OUT 49149,16
60 OUT 65533,12: OUT 49149,18
70 OUT 65533,13: OUT 49149,0

```

© Командин Илья, г. Калуга.

КОРР: Предлагаю пару процедур проявления экрана.

```

;      "APPERNCE BY PLOT"
;      (C) ZX-MANIAC, 1995
;      -----
;
;      ORG      50000
;      ENT     $
ADDR   EQU     40000
;
M1     LD      B,10
;
;      LD      HL,16384
;      LD      DE,ADDR
M2     LD      A,R
;      EX     DE,HL
;      AND    (HL)
;      EX     DE,HL
;      OR     (HL)
;      LD     (HL),A
;      INC    HL
;      INC    DE
;      LD     A,H
;      CP     88
;      JR     NZ,M2
;      DJNZ   M1
;      LD     HL,ADDR
;      LD     DE,16384
;      LD     BC,6912
;      LDIR
;      RET

```

Длина процедуры 36 байт.

Данная процедура проявляет картинку, загруженную по адресу ADDR, по точкам в случайном порядке, используя регистр регенерации.

Следующая процедура действует так: по экрану пробегает квадрат желтого цвета, оставляя за собой проявленную картинку. Эффект достигается за счет манипуляции атрибутами. Скорость работы программы может регулироваться в широких пределах путем изменения числа в метке "PAUSE".

```
;      "APPERNCE BY PLOT"
;      (C) ZX-MANIAC, 1995
;      -----
      ORG      50000
      ENT      $
ADDR   EQU     40000
PAUSE  EQU     1300
      LD      HL,22528
      LD      DE,22529
      LD      BC,767
      LD      (HL),0
      LDIR
      LD      HL,ADDR
      LD      DE,16384
      LD      BC,6144
      LDIR
      LD      BC,768
LOOP   LD      A,54
      LD      (DE),A
      PUSH   HL
      LD      HL,PAUSE
WAIT   DEC     HL
      LD      A,H
      OR     L
      JR     NZ,WAIT
      POP    HL
      LD      A,(HL)
      LD      (DE),A
      INC    HL
      INC    DE
      DEC    BC
      LD      A,B
      OR     C
      JR     NZ,LOOP
      RET
```

Длина процедуры 50 байт.

Обе процедуры лучше всего смотрятся при темном PAPER и светлом INK.

ИФК: свое решение задачи, заданной Сергеем Колотовым прислал нам Антон Сидоренко (Logic games) из п.Мурмаши, Мурманской области:

КОРР: Хочу предложить вам процедуру-календарь. Сделать ее предложил Сергей Колотов. Правда, он предложил не только это, но календарь - единственная понравившаяся мне задача.

Вот мой вариант: DE - год, C - месяц;

```
LD      A,C
ADD     A,248
SBC     A,248
OR      30
DEC     C
DEC     C
RET     NZ
LD      A,E
AND     252
CP      E
LD      A,29
SBC     A,C
RET
```

Длина процедуры 18 байт.

ИФК: А сейчас предоставим слово Котлярову Евгению из г. Оренбурга.

КОРР: Я хочу предложить читателям две свои процедуры в машинных кодах и надеюсь, что они будут полезны.

Первая процедура предназначена для поиска шрифтов и спрайтов. Подобные программы на БЕЙСИКЕ уже печатались в ZX РЕВИЮ, но эта отличается большими возможностями: она компактна (239 байт), релоцируема, с ее помощью можно искать не только шрифты, но и спрайты размером от 1*1 до 7*31 знакомест. Самое удобное для нее место, на мой взгляд, - буфер принтера.

Клавиши управления:

- 1 - уменьшить длину окна;
- 2 - увеличить длину окна;
- 3 - увеличить высоту окна;
- 4 - уменьшить высоту окна;
- 5 - переключение спрайт-шрифт;
- 6 - увеличение адреса на 32;
- 7 - уменьшение адреса на 32;
- 8 - увеличение адреса на 1;
- 9 - уменьшение адреса на 1;
- 0 - выход из программы.

Вызвать программу можно, например, так:

```
RANDOMIZE ADDRESS1: LET A=USR ADDRESS2,
```

где ADDRESS1 - адрес начала поиска, ADDRESS2 - адрес загрузки. После выхода в А, будет возвращен текущий адрес. Определить режим поиска можно по переменной FLAG, находящейся по адресу #59FD в области атрибутов. Если это знакоместо черного цвета, значит производится поиск шрифтов, если яркого - спрайтов.

Вот ее листинг:

```
SEED      EQU      23670
HEI       EQU      #59FF
LEN       EQU      #59FE
FLAG      EQU      #59FD
          ORG      40000
          ENT      $
          LD       A, 2
          LD       (LEN), A
          LD       (HEI), A
          CALL    #1601
          XOR      A
          LD       (FLAG), A
          LD       HL, (SEED)
START     PUSH    HL
          PUSH    HL
          LD       BC, #0921
          CALL    #0DD9
          POP     BC
          CALL    #2D2B
          CALL    #2DE3
          POP     HL
          LD       A, 32
          RST    16
          LD       A, #EF
          IN      A, (#FE)
          RRA
          LD       B, H
          LD       C, L
          RET     NC
          RRA
          JR      C, L2
          DEC     HL
L2        RRA
          JR      C, L3
          INC     HL
L3        RRA
```

	LD	BC, 32
	JR	C, L4
	SBC	HL, BC
L4	RRA	
	JR	C, L5
	ADD	HL, BC
L5	LD	A, #F7
	IN	A, (#FE)
	LD	C, A
	RR	C
	LD	A, (LEN)
	JR	C, L6
	DEC	A
	JR	NZ, L6
	INC	A
L6	RR	C
	JR	C, L70
	CP	31
	JR	Z, L7
	INC	A
L70	LD	(LEN), A
L7	RR	C
	LD	A, (HEI)
	JR	C, L8
	CP	8
	JR	Z, L8
	INC	A
L8	RR	C
	JR	C, L90
	DEC	A
	JR	Z, L9
	PUSH	AF
	PUSH	BC
	PUSH	HL
	XOR	7
	INC	A
	LD	B, A
	CALL	#0E44
	POP	HL
	POP	BC
	POP	AF
L90	LD	(HEI), A
L9	RR	C
	LD	A, (FLAG)
	JR	C, RUN
	CPL	
	LD	(FLAG), A
RUN	OR	A
	JR	Z, FO_FIND
	JR	FIND
START1	JR	START
FIND	PUSH	HL
	LD	BC, (LEN)
	LD	DE, #5000
L10	PUSH	BC
	LD	B, 8
L11	PUSH	BC
	LD	B, 0
	LDIR	
	XOR	A
	LD	(DE), A
	POP	BC
	INC	D
	LD	A, E

```

SUB      C
LD       E, A
DJNZ    L11
POP      BC
LD       D, %01010000
ADD     A, 32
LD       E, A
DJNZ    L10
POP      HL
JR       START1
FO_FIND PUSH  HL
LD       A, (LEN)
LD       B, A
LD       DE, #5000
L12     PUSH  BC
        PUSH  DE
LD       A, (HEI)
        RLCA
        RLCA
        RLCA
L13     LD      C, A
        LDI
LD       A, B
CP       1
JR       NZ, L14
XOR     A
L14     LD      (DE), A
        DEC  DE
LD       A, D
AND     7
CP       7
JR       NZ, L15
LD       A, D
AND     %11111000
LD       D, A
LD       A, E
ADD     A, 32
LD       E, A
L15     JR      L16
L16     INC  D
LD       A, C
OR      A
JR       NZ, L13
POP      DE
INC     DE
POP      BC
DJNZ    L12
POP      HL
JR       START1

```

Вторая процедура реализует оригинальную бегущую строку. С ее помощью можно выводить текст различными модифицированными шрифтами (наклонным, утолщенным) и их комбинациями. Причем вид шрифта задается всего 8 байтами. После метки TABLE я привел несколько примеров. Для того чтобы просмотреть их, нужно в строке LD DE, TABLE+16 вместо 16 подставить какое-либо другое число, кратное 8. Я думаю, после просмотра демонстрации, принцип работы процедуры станет, ясен, и читатели сами смогут придумать другие модификации шрифтов.

Листинг процедуры:

```

ORG     30000
ENT     $
CALL    INIT
START   HALT
LD      B, 2
LOOP    PUSH  BC
        CALL  STROKA

```

```

POP      BC
DJNZ    LOOP
LD      A, #7F
IN      A, (#FE)
RRA
JR      C, START
RET
INIT    LD      HL, TEXT
        LD      (STARTL), HL
        LD      (COUNTL), HL
        LD      A, 1
        LD      (POSLET), A
        RET
STROKA LD      A, (POSLET)
        RRCA
        LD      (POSLET), A
        JR      NZ, BEGIN
        LD      HL, (COUNTL)
        LD      A, (HL)
        OR      A
        JR      NZ, L1
        LD      HL, (STARTL)
        LD      A, (HL)
L1      INC     HL
        LD      (COUNTL), HL
        LD      L, A
        LD      H, 0
        ADD     HL, HL
        ADD     HL, HL
        ADD     HL, HL
        LD      DE, (23606)
        ADD     HL, DE
        LD      (ADRES), HL
BEGIN   LD      DE, (ADRES)
        EXX
        LD      DE, TABLE+16
        EXX
        LD      BC, (POSLET)
        LD      HL, #401F
L2      LD      A, (DE)
        AND     C
        EXX
        JR      Z, L3
        LD      A, (DE)
L3      INC     DE
        EXX
        OR      (HL)
        LD      (HL), A
        INC     H
        INC     DE
        DJNZ    L2
        LD      HL, #401F
        LD      A, 8
L41     PUSH    HL
        LD      B, 8
L4      RL      (HL)
        DEC     L
        DJNZ    L4

```

```

        POP     HL
        INC    H
        DEC    A
        OR     A
        JR     NZ,L41
        RET
; *****
POSLET  DEFW   #0800
COUNTL DEFW   0
STARTL  DEFW   0
ADRES   DEFW   0
TEXT    DEFM   "NEW RUNNING LINE "
        DEFM   "(C)1996 KOTLYAROV E.S."
        NOP
TABLE   DEFB   6,6,6,6,6,6,6,6
        DEFB   3,6,12,24,48,96,192,128
        DEFB   2,2,2,2,6,6,6,6
        DEFB   6,6,3,3,3,6,6,6
        DEFB   3,3,3,3,6,6,6,6
        DEFB   6,6,6,7,5,5,5,7
        DEFB   6,6,6,4,4,6,6,6
        DEFB   3,3,6,6,6,6,12,12
*****

```

© А. Шокало, г. Тернополь.

В ZX 95/4, в этуодах, прочитал статью о процедурах KEY-INPUT. Хорошее начинание. Алексей Артамонов предложил свою процедуру опроса клавиатуры. Сейчас я работаю над одной небольшой игрушкой, и там я применил такой же метод, хотя сама процедура у меня больше. Вы говорите, что эта программа "явно не удовлетворяет требованиям, предъявляемым к такого рода процедурам". Так вот можно сделать по другому. По очереди запуская процедуры и накладывая результаты друг на друга (OR). Но зачем делать управление от нескольких интерфейсов. Ведь в таком варианте процедуры может быть случай, когда указатель должен передвинуться вверх, вниз, влево и вправо одновременно, и при этом нажата клавиша огонь.

А сейчас, в продолжение данной темы, я хочу добавить процедуру, которую можно использовать совместно с процедурой Алексея. Она предназначена для выбора клавиш. Ет я тоже использую в своей игре.

Немного о самой программе. Точка входа REDEF. Никаких входных данных не требуется. После запуска программа ожидает нажатия на любую клавишу. После выхода из процедуры в регистрах: H - старший байт адреса порта, L - маска бита нажатой кнопки.

```

; Redefine keys (C) FREE Group 1995
REDEF   LD     BC,#7FFE
KEY_L   IN     A,(C)
        AND    %11111
        CP     %11111
        JR     NZ,KEY_PR
        SCF
        RR     B
        JR     C,KEY_L
        JR     REDEF
KEY_PR  XOR    %11111
        LD     L,A
        LD     H,B
        RET

```

Но если Вы предлагаете пользователю выбрать клавиши, то нужно показывать, какую именно он нажал. Поэтому у меня есть процедура, которая, по выходным данным предыдущей, в регистре A выдает код символа на данной клавише.

```

TRANSF CALL COUNN
        LD     E,A
        LD     A,4
        SUB   E
        LD     E,A
        LD     A,H
        XOR   255
        CALL  COUNN

```

```

LD      D,A
LD      A,7
SUB     D
SLA     E
SLA     E
SLA     E
ADD     A,E
LD      E,A
LD      D,0
LD      HL,#205
ADD     HL,DE
LD      A,(HL)
RET
COUNN   PUSH   BC
LD      B,0
COU_L   RRA
JR      C,COU_E
INC     B
JR      COU_L
COU_E   LD      A,B
POP     BC
RET

```

Данная программа просто делает необходимые пересчеты, и по таблице в ПЗУ выбирает код нажатой клавиши.

© А.Волков, г. Березники, Пермской обл.

Хочу предложить в рубрику "Этюды" свою программу. Подобной я у вас еще не встречал. Эта программа выполняет гашение экрана случайным образом по знакам.

```

10      ORG     40000
20      ENT     $
30 L1    LD     BC,3
40      LD     HL,22528
50      LD     E,B
60 L2    LD     A,(HL)
70      AND    A
80      JR     Z,L4
90      EXX
100     INC    DE
110     LD     A,D
120     AND    7
130     OR     0
140     LD     D,A
150     LD     A,(DE)
160     EXX
170     AND    7
180     LD     D,A
190     LD     A,(HL)
200     AND    7
210     SUB    D
220     JR     NC,L3
230     XOR    A
240 L3   LD     (HL),A
250     LD     E,1
260 L4   INC    HL
270     DJNZ  L2
280     DEC    C
290     JR     NZ,L2
300     LD     BC,10000
310     CALL  L5
320     DEC    E
330     RET    NZ
340     JR     L1
350 L5   PUSH  BC

```

```

360     PUSH AF
370 L6   DEC     BC
380     LD     A,B
390     OR     C
400     JR     NZ,L6
410     POP   AF
420     POP   BC
430     RET

```

Длина процедуры 60 байт. В 300 строке можно изменять скорость выполнения программы в широких пределах. Для испытания программы можно использовать следующую программу на BASICe.

```

10 CLEAR 39999
20 LOAD "CLS" CODE 40000
30 LOAD "SCREEN" SCREEN$
40 RANDOMIZE USR 40000

```

В 30 строке загружается любой файл-экран.

© Прокопенко А.В.

Предлагаю релоцируемую программу LOAD-MERGE, реализующую загрузку с диска BASIC-программ без автостарта. Пользоваться программой очень просто: наберите строку, что-то вроде:

```
PRINT USR ADR : REM NAME,
```

где ADR - адрес загружаемой программы, NAME - имя загружаемой BASIC-программы. Если файла, указанного после REM нет на диске, то будет выдано сообщение: "Invalid file name", иначе загрузка завершится с сообщением "OK" с номером стартующей строки.

Программа LOAD-MERGE

```

START   ORG     30000
        LD     C,#18           ; Настройка на диск.
        CALL  #3D13
        LD     HL,#5CDD       ; Заполнение имя файла пробелами.
        PUSH  HL
        LD     B,8
ZAP_    LD     (HL),#20
        INC   HL
        DJNZ  ZAP_
        LD     (HL),"B"       ; Установка некоторых
        LD     L,#06         ; переменных DOS.
        LD     (HL),#09
        LD     L,#F9
        LD     (HL),#00
        LD     HL,(#5C5D);Системная переменная CH_ADD.
        INC   HL
        LD     B,8           ; Переброска не более 8 байт,
        POP   DE             ; записанных за REM в дескриптор
IN_F    INC   HL             ; файла.
        LD     A,(HL)
        CP    #0D            ; Если конец строки - то и конец
        JR    Z,IN_F1       ; файла.
        LD     (DE),A
        INC   DE
        DJNZ  IN_F
IN_F1   LD     C,#0A         ; Проверка на наличие
        CALL  #3D13         ; файла на диске.
        INC   C
        JR    NZ,LOAD       ; Если файл есть - то переход на LOAD,
        RST  #08            ; иначе сообщение об ошибке.
        DEFB  #0E
LOAD    XOR   A             ; По адресу #5DA0 должно быть #00.
        LD     (#5DA0),A
        LD     C,#0E         ; Загрузка файла.
        CALL  #3D13
        LD     HL,(#5C42)    ; В HL номер стартующей строки.
        LD     (#5C45),HL    ; Номер выполняющейся строки.
        RST  #08            ; Сообщение "OK"
        DEFB  #FF

```

© М. Тамаринов, г. Соликамск.

Хочу предложить вашему вниманию две процедуры. Первая – компрессор символьной информации. Он был написан мной специально для компрессии игрового поля.

Его достоинства:

1. Не важно сколько раз повторяется байт.
2. Компрессия на выходе будет либо меньше, чем несжатые данные (если есть что сжимать), либо равна (если сжимать нечего), но никак не больше.

Его недостатки:

1. В данных для компрессии не должны встречаться байты #FE и #FF (254, 255). Поэтому компрессор годен для сжатия символьной информации.

Компрессор:

```
SMARTCOMP LD IX, (BUFF)
           LD HL, (DEST)
           LD D, 1
LOOP      LD A, (IX)
           CP #FF
           JR Z, END
           CP (IX+1)
           JR NZ, FORM
           LD A, 0
           CP #FF
           JR NC, OVELOW
PASS      INC D
           INC IX
           JR LOOP
FORM      LD A, D
           CP 3
           JR C, SINGLE
           LD (HL), #FE
           INC HL
           LD (HL), D
           INC HL
           LD A, (IX)
           LD (HL), A
           INC HL
           LD D, 0
           JR PASS
SINGLE    LD A, (IX)
FILL     LD (HL), A
           INC HL
           DEC D
           JR NZ, FULL
           JR PASS
OVFLOW   LD (HL), #FE
           INC HL
           LD (HL), D
           INC HL
           LD A, (IX)
           LD (HL), A
           INC HL
           INC IX
           LD D, 0
           JR LOOP
END      LD (HL), #FF
           LD B, H
           LD C, L
           RET
```

BUFF - адрес, с которого начинаются данные для компрессора.

DEST - адрес, куда нужно поместить сжатые данные.

Декомпрессор:

```
LD IX, (DEST)
```

```

LD      HL, (BUFF)
DEL1   LD      A, (IX)
       CP      #FF
       RET     Z
       CP      #FE
       JR      NZ, ONE
       LD      B, (IX+1)
       LD      A, (IX+2)
DEFLAT LD      (HL), A
       INC     HL
       DJNZ   DEFLAT
       INC     IX
       INC     IX
       INC     IX
       JR      DFL1
       LD      (HL), A
       INC     IX
       INC     HL
       JR      DFL1

```

Вторая процедура была написана мною в дополнение к статье Д. Баянова:

```

; быстрая печать шаблона 8*8
; на входе: BC - координаты; A - код символа
;

```

```

WRITE  DI
       LD      H, 0
       LD      L, A
       ADD     HL, HL
       ADD     HL, HL
       ADD     HL, HL
       LD      DE, BASE      ; Адрес фонта
       ADD     HL, DE
       LD      (STASC+1), HL
       LD      (STC), SP
       LD      A, C
       AND     #18
       ADD     A, #40        ; При использовании
                           ; этой и других
       LD      H, A          ; процедур, использующих стек,
                           ; необходимо отключать прерывания,
       LD      A, C          ; иначе при первом же прерывании,
                           ; будут испорчены данные.
       RCA
                           ;
       RCA
                           ;
       ADD     A, B          ;
                           ;
       LD      L, A
STASC  LD      SP, 0
       POP     DE
       LD      (HL), E
       INC     H
       LD      (HL), D
       INC     H
       POP     DE
       LD      (HL), E
       INC     H
       LD      (HL), D
       INC     H
       POP     DE
       LD      (HL), E
       INC     H
       LD      (HL), D
       INC     H
       POP     DE
       LD      (HL), D
       LD      SP, (STC)
       EI

```

```
STC      DEFW      0
BASE     ...
```

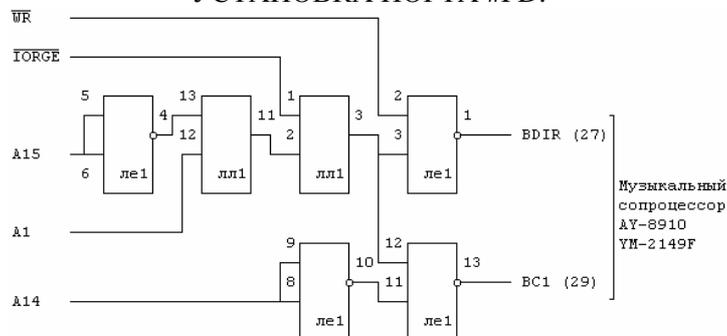
ФОРУМ



ИФК: Великая слава SCORPION'a распространилась уже во все уголки нашей страны. И такой ее недостаток (а скорее достоинство) как полная дешифрация портов не останавливает, а лишь подстегивает их обладателей на новые поиски и открытия. На наш адрес пришло много писем, в которых вы, наши читатели, пытаетесь устранить несовместимость некоторых программ с этим, в остальном замечательным, компьютером. Все доработки можно разделить на программные и аппаратные. Мы приведем оба варианта. Итак, вот что нам пишут:

© Юра Васин, Ефим Тувиков. SCORPION CLUB.

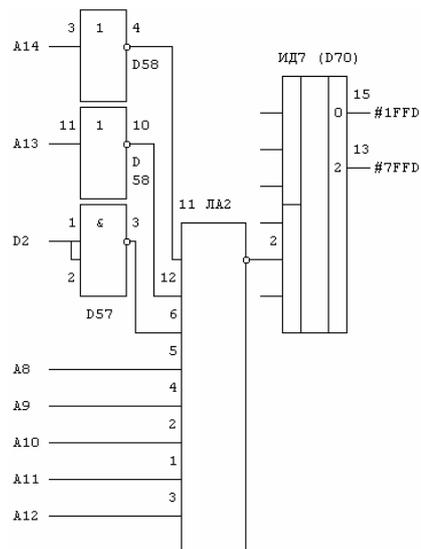
УСТАНОВКА ПОРТА #FD.



КОРР: При установке этих микросхем на компьютер появляется порт #FD, и это говорит о том, что начинают работать все цифровые музыкальные программы. Схема состоит из 2-х микросхем: K555ЛЕ1 и K555ЛЛ1.

WARNING: Не забудьте подпаять к 7-ым выводам обоих микросхем землю, а к 14-ым - +5В. Ножки у АТ -8910 или YM2149F с номерами 27,29 (они являются BDIR,BC1) надо отрезать от платы и подать на них сигналы с нашей схемы.

При "глючной" работе компьютера с нашей схемой (что имеет очень малый процент вероятности) следует сменить сигнал IORGE на IORQ.



© Ермишин Алексей, Мигунов Анатолий.

ЖЕСТКАЯ ДЕШИФРАЦИЯ ПОРТА #1FFD В ПК "SKORPION ZS-256".

КОРР: Предлагаю вашему вниманию доработку Scorpion'a, после которой, будут работать все программы с переключением страниц по #FD.

Кстати сказать, не нужна для этого никакая специальная кнопка ON/OFF, как на PROF1. Просто-напросто необходимо исключить нежелательное обращение к порту #1FFD. Что, собственно, и делает сия доработка, которую придумал и воплотил в жизнь ЕРЕМИШИН АЛЕКСЕЙ (ROMA).

А теперь более подробно на словах. Для доработки понадобятся микросхемы K555ЛA2 и K555ИД7 (можно использовать серию 1533), а также сопротивление 10 кОм. Порядок действия таков:

1. У м/с D53 (ИД7) отрезать от платы 6 ножку, затем отрезать от корпуса (не от платы) ножки 15 и 13. Новую м/с ИД7 (условно D70) повесить сверху на D53, спаяв у них между собой ножки 1,3,4,5,6,8,16. Ноги 15 и 13 (D70) припаять к отрезанным ногам м/с D53 соответственно.

2. На 2 ножку D70 подать сигнал с 8 ножки ЛА2; ЛА2 повесить на D15 (КП11), припаяв ножки 3,7,14 (ЛА2) к 3,8,16 (D15) соответственно.
3. В схеме задействованы неиспользуемые элементы микросхем D57 (ЛА3) и D58 (ЛН1). Кстати, убедитесь, что эти элементы у Вас действительно свободны, иначе ищите подобный элемент, либо можно повесить сверху аналогичную микросхему.
4. Остальные сигналы берутся с платы компьютера. Все они отходят от ПЗУ, но следите за тем, чтобы соединяющие проводники были минимальной длины.
5. Спаянные 6-ые ножки м/с D53 и D70 подпаять к +5В через сопротивление 10 кОм, чтобы там постоянно сидел уровень логической "1". Это необходимо для того, чтобы слушать оцифрованные мелодии.

Вот и вся доработка. После получаса работы Вы можете насладиться, не идущими доселе музыкалками и прочими программами отечественных хакеров и программистов. Такими как STS, MIRACLE 128, SIMPSONS 128, INSULT, и др.

Теперь SCORPION будет работать так, как надо, и даже такое действие: LD A,#1F: OUT (#FD),A не приведет к зависанию компьютера, а включится страница. Чтобы обратиться к порту #1FFD, необходимо его полностью указать в регистре BC.

Примечание: в приведенной выше доработке используется D2 порта #1FFD. Поэтому, из-за отсутствия информации по турбированию SCORPION'a (возможно, там тоже используется именно этот разряд), или если в вашем компьютере этот разряд используется для каких-либо личных целей, то в этом случае из схемы надо исключить элемент D57, и на 6 ногу ЛА2 подать +5В через сопротивление 10 кОм. Но при этом некоторые программы могут не работать, в частности "INSULT MEGADEMO v1.3".

Доработка опробована на трех SCORPION'ах - никаких проблем. Теперь с уверенностью можно сказать, что на SCORPION'e работает 99.99 % спектрумовского программного обеспечения.

В данное время ROMA работает над расширением SCORPION'a до 512 кВ (прикиньте возможности программы "MagOS" и "IS-DOS"). Эта доработка сделана частично, т.к. дополнительные страницы не переключаются из теневого монитора (компьютер виснет), но написан драйвер RAM-диска для IS-DOS, который использует все 512 кВ, и при этом работающий нормально. По поводу зависания стоит вопрос: дело в железе, или же в ПЗУ? Для расширения был использован разряд D порта #1FFD. Если кто-то решил проблему переключения дополнительных страниц из теневого монитора, то просьба откликнуться.

© А.Соловьев, г. Андреаполь.

ДОРАБОТКА STS 3.3.

Хотим сообщить вам, что мы успешно переделали STS 3.3, который не работал из-за неполной дешифрации #7FFD. Итак, начнем:

CLEAR 24499 - устанавливаем RAMTOR.

RAND.USR 15619:REM:LOAD "sts 3.3" CODE - загружаем.

POKE 65299,88 - обеспечиваем нормальную инсталляцию в 10 страницу.

POKE 65366,80 - обеспечиваем нормальную инсталляцию в любую страницу.

POKE 58922,84 - нормальный выход в TASM.

POKE 58915,80 - нормальный возврат в BASIC.

POKE 58894,80 - нормальный выход в TR-DOS.

RAND.USR 15619:REM:ERAZE "sts 3.3" CODE

RAND.USR 15619:REM:SAVE "sts 3.3" CODE 56064,9472

Остается добавить только то, что при первоначальном входе в STS Вам придется изменить параметр BANK, устанавливая его не в 18...1F, а в 58...5F, то есть:

58 - нулевая страница

59 - первая страница

...

...

...

5F - седьмая страница.

© М. Торгунов, г. Красноярск.

ДОРАБОТКА "TASM 128".

Цель этого письма - помочь владельцам "Скорпионов", "АТМ" и др. Прочитав раздел "FORUM" в ZX РЕВЮ 95/4, где пишет Павел Ляховский, я решил помочь ему и всем остальным, у которых не работает "TASM-128".

Вообще существует этот пакет, который работает на "Скорпионе". Он иногда "зашивается" в ROM-диск этого компьютера. Теперь ближе к делу:

Программа TASM 3.0 не работает, потому, что обращение к порту #7FFD идет как обращение к порту #FD.

Чтобы изменить программу TASM 3.0, нужно загрузить кодовый блок TASM 3.0 - под адрес 49152, далее изменить следующие ячейки памяти:

```
#C019=#C3
#C01A=#40 ] JP 23872
#C01B=#5D ]
#C022=#C3
#C023=#50 ] JP 23888
#C024=#5D ]
```

Далее выгрузите программу - SAVE "TASM 3.2" CODE 49152, 10448

Если Вы не производили изменений, о которых написано в ZX РЕВЮ 95/5, то их сделать тоже необходимо для работы с TASM'ом.

Далее необходим специальный загрузчик.

1. Наберите строку : 255 REM 34-пробела.
2. В сформированной строке отасSEMBлируйте программу с адреса 23872:

```
LD (23904),BC
LD BC,#7FFD
OUT (C),A
LD BC,(23904)
JP #C000
LD (23904),BC
LD BC,#7FFD
OUT (C),A
LD BC,(23904)
JP 49190
DEFW #FFFF
```

3. Объедините по MERGE в TR-DOS'e эту строку с загрузчиком пакета - TRACK 2&3.
4. Сотрите все строки, кроме нулевой и 255-ой.
5. Дайте РОКЕ 23952,0.

И осталось набрать бейсик-строки следующего содержания:

```
10 POKE VAL "23693",VAL "7": POKE VAL "23624",VAL "7": OUT CODE "RETURN",NOT PI: OUT
VAL "32765",VAL "20": RANDOMIZE USR VAL "15619": REM : LOAD "TASM3.2"CODE VAL
"49152",VAL "10448"
20 OUT VAL "32765",VAL "20": RANDOMIZE USR VAL "49152": STOP
```

Для владельцев "Скорпионов" посоветую еще набрать следующую строку:

```
30 POKE 16384,243: POKE 16385,118: POKE 16386,251: POKE 16387,201: RANDOMIZE USR
16384
```

Эта строка обеспечивает вход в SHADOW MONITOR. Даете RUN 30, нажимаете MAGIC, отлаживаете свою программу и, по команде JP C в мониторе или "CONTINUE PROGRAM", выходите в BASIC.

Теперь загрузчик Вы написали, можете его выгружать: SAVE "TASM 3.2". Вот и все. После этих изменений ассемблер работает на любых компьютерах.

Приведенный мною фрагмент занимается коммутацией страниц при подаче команд: O,Q,R,S. Он раньше располагался с адреса 23597, и выглядел так:

```
L1 JP 49152 (49190)
LD A,20
L2 OUT (253),A
JR L1
```

Обращение к порту #FD вынужденное, иначе этот резидент "залезает" в важные области системных переменных. Я, в свою очередь, заменил команду OUT (253),A и кусочек JR L1 на JP 23872 (23888), где и происходит корректное переключение порта #7FFD.

После всех выше перечисленных изменений программа работает просто великолепно!!!

© О. Литвинов.

ДОРАБОТКА ПОСЕКТОРНЫХ ЗАГРУЗЧИКОВ.

Часто при работе посекторных загрузчиков, использующих функцию 5 TR-DOS и точку входа #3D13, дисковод "спотыкается". Вместо того, чтобы начать чтение нового блока с уже найденной дорожки, головка дисковода начинает судорожно что-то искать, затем откатывается к началу диска, потом заново ус-

танавливается на ту же дорожку и, наконец, приступает к чтению. Очевидно, происходит какой-то сбой, связанный с потерей данных. Загрузчики, использующие точку входа в DOS #3D2F, такого себе не позволяют, а при загрузке из BASIC'a головка каждый раз "ездит" на нулевую дорожку, чтобы прочесть заголовок, так что ошибки позиционирования не происходит.

Исправить досадный недостаток в работе посекторного загрузчика просто. Зная, что в данном месте программы происходит такая неприятность (а раз уж она случилась, значит, будет повторяться), добавьте 12 байт:

```
LD     DE, (#5CF4)    ;Номер дорожки и сектора.
DEC    D              ;Или INC D, переход на
LD     BC, 5          ;другую дорожку.
CALL   #3D13         ;Чтение заголовка сектора.
LD     DE, (#5CF4)    ;
INC    D              ;Или DEC D, восстановление номера
                          ;дорожки.
                          ;-----
                          ; ЧТЕНИЕ БЛОКА
```

Можно еще сократить эту добавку, сохраняя содержимое регистровой пары DE на стеке.

Еще я хочу сказать о профзабывчивости, связанной с компьютерами. Интересующая читателей "ZX-РЕВЮ" разновидность его называется так:

Звездная болезнь "крутизны" в синклеризме.

Главный симптом этого заболевания состоит в том, что больной, устремляя взор в небеса, перестает смотреть под ноги. Один из интересных случаев связан с книгой Ларченко и Родионова "ZX Spectrum & TR-DOS для пользователей и программистов".

В драйвере дисковых операций допущена грубая ошибка – в функции 2 (позиционирование на заданный логический трек) неверно рассчитывается номер физической дорожки. Привожу фрагмент программы:

```
#F88A   SLA    D
        BIT    7, E
        JR     Z, #F895
        BIT    1, D
        JR     Z, #F895
        DEC    D
#F895   BIT    7, E
        JR     Z, #F8AA
#F899   BIT    7, (HL)
        JR     Z, #F882
        SRL    D
```

Итак, что мы видим? Содержимое регистра D (номер логической дорожки) умножается на 2 вместо того, чтобы делиться. В результате происходит следующее: Вы задаете номер логического трека, скажем, 50 или 51, и дисковод усердно принимается искать физическую дорожку с номером 100 (в обоих случаях). Естественно, это не удается, и головка устанавливается на последнюю существующую дорожку. Если дорожка с искомым номером все же существует, выбирается всегда головка 0.

Терапия (для программы): по адресу #F88A поставить JR #F899 (байты #18 и #0D). Следующие 9 байт можно обнулить. После этого номер логического трека, как и положено, делится на 2 (см. последнюю строку листинга). Функция 2 (и все остальные) после внесенных изменений на моем дисководе (80 дорожек, 2 головки) работают нормально. Не поручусь, что для других типов дисководов дела будут обстоять так же.

Все бы ничего, подумаешь - часто ли она нужна, эта опция? Жаль только, если исправление этой ошибки не вошло в число тех "несущественных изменений", с которыми драйвер "вошел в базовое программное обеспечение компьютера "Scorpion ZS256"" (с.212), то есть, скорее всего, был "зашит" в ПЗУ.

Вторая история болезни связана с той же книгой. На с.181 читаем: "нулевой логической дорожке соответствует нулевая нижняя физическая, первой логической - нулевая верхняя физическая". На с. 202 узнаем, что занесенный в бит 4 системного регистра "0" соответствует первой магнитной головке или нижней стороне дискеты, 1 - второй магнитной головке или верхней стороне". Логично? Вполне! Тем более, что создатели ADS даже позаботились изобразить дискету, чтобы ни у кого не возникло сомнений, что нулевая сторона - нижняя, а первая - верхняя.

Я, помнится, пару дней ломал голову над вопросом, почему же, в таком случае, при чтении с нечетной логической дорожки в бит D4 системного регистра засылается 0, а при чтении с четной дорожки - 1 (См., например, "ZX-РЕВЮ", 1994, N4, с.9.). Кто же неправ? Некоторое облегчение принесла брошюра Ю.

Поморцева "TR-DOS для профессионалов и любителей", присланная "ИНФОРКОМОМ". Я узнал, что есть, по крайней мере, еще один человек, который, как и я, считает, что нулевая сторона дискеты - верхняя, и логические дорожки с четными номерами находятся именно на верхней, нулевой стороне.

Несмотря ни на что, у программистов болезнь протекает в мягкой, SOFT - форме. Наверное, еще ни один пользовательский "Спектр" не сгорел из-за ошибки в программе. Оценить тяжесть заболевания у "жестянщиков", присылающих в журнал схемы с несущественными изменениями вроде перепутанных выводов или пропущенных буферных элементов не берусь - пусть это сделают те, кто эти схемы спаял.

Господа гении! Вы очень умные, но все же вряд ли создаете свои творения за пять дней, как господь бог. Так будьте добры, потратьте еще несколько дней на проверку вашего детища! Помните: если вам кажется, что все хорошо и правильно, но вы не проверили, значит, ошибка есть.

© Колотов Сергей, 1996.

НЕСТАНДАРТНЫЕ МАГНИТОФОННЫЕ ЗАГРУЗЧИКИ.

Пишу вот по какому поводу. Все началось с того, что я слишком внимательно читал 6-ой номер ZX-ревью за 1994 год и в разделе "Читатель-читателю" наткнулся на статью "Нестандартные загрузчики". Появилась идея создать универсальную процедуру загрузки программ с магнитофона эффективным и оригинальным образом.

ИФК: Большинство читателей, которые пользуются дисковыми, возможно, сталкивались с программами, которые в процессе загрузки еще и выводят текст на экран, проигрывают музыку и т.п. К таким можно отнести BATTLE COMMAND и INSULT MEGADEMO. Так вот, Сергей предлагает дать возможность использовать такие эффекты при загрузке с магнитофона. Но дадим слово автору.

При использовании метода, описанного ниже, можно во время загрузки блока кодов выполнять любую программу пользователя (естественно, она будет выполняться медленнее). Например, загружая игру, можно одновременно читать ее описание, которое постепенно будет выводиться на экран; можно грузить части программы в любое место памяти, тут же их декодировать и запускать (конечно, они должны вызывать процедуру загрузки). И это в одном неразрывном кодовом блоке.

Процедура загрузки файлов с магнитофона с
параллельной работой программы пользователя

Written by Kolotow Sergey

Copyright SerzhSoft from Shadrinsk

```
10      ORG      30000
;Считывание первого встреченного кодового файла с заголовком
20 FILE  LD      IX,HEADER      ;адрес загрузки заголовка файла
30      LD      DE,17          ;длина заголовка
40      XOR     A              ;загружаться будет заголовок
50      SCF                    ;выполнять чтение, а не проверку файла
60      CALL   1366            ;вызов процедуры чтения в ПЗУ
70      JR     NC,FILE         ;если возникла ошибка, то повторить
80      LD     A,(HEADER)      ;взяли первый байт считанного заголовка
90      CP     3               ;это блок кодов?
100     JR     NZ,FILE         ;если нет, повторить загрузку
110     LD     IX,(START)      ;адрес загрузки блока кодов
120     LD     DE,(LENGTH)     ;длина блока кодов
130     LD     A,255           ;загружаем файл
140     SCF                    ;загрузка, а не проверка
;Процедура загрузки, аналогичная CALL 2050
150 LOADER CALL PROC          ;вызвать процедуру загрузки
160     CALL  #053F           ;восстановить бордюр и, если нажата
;клавиша BREAK, то выдать сообщение об этом
170     RET     C              ;выход, если нет ошибки
180     RST    8               ;иначе - печать сообщения
190     DEFB   26              ; "Tape loading error"
;Главная процедура загрузки
;На входе: IX - адрес начала загрузки файла
;          DE - длина загружаемого файла
;          A - флаг файл/заголовок:
;              #00 - заголовок
;              #FF - файл
;          флаг C=1 - если загрузка (команда LOAD)
;          флаг C=0 - если проверка (команда VERIFY)
```

```

200 PROC DI ;запретить прерывания
210 DEC DE ;уменьшить длину файла на 1
220 LD (CNTLEN),DE ;и запомнить полученное значение
230 LD DE,1 ;будем читать один байт
240 INC D ;подготовка регистров, как и в ПЗУ
250 EX AF,AF'
260 DEC D
270 CALL #0562 ;загрузка байта процедурой ПЗУ
280 LD A,127 ;проверяем нажата ли клавиша SPACE
290 IN A,(254)
300 RRA
310 RET NC ;если да, то аварийный выход
320 LD A,E ;первый байт считался нормально ?
330 CP 1
340 RET NC ;если нет, то выход с ошибкой
350 EXX ;переключили набор регистров
360 PUSH HL ;сохранили HL' для нормального завершения
370 LD (MEM_SP),SP ;запомнили указатель стека
380 CALL MYPROC ;вызов процедуры пользователя
390 LPLOAD CALL LOAD1B ;продолжать загрузку до конца файла по
400 JR LPLOAD ;одному байту
;Процедура считывания одного байта
410 LOAD1B PUSH AF ;сохранили AF на стеке
420 EXX ;поменяли набор регистров
430 LD E,1 ;загружать один байт
440 CALL #05A9 ;процедура загрузки из ПЗУ
450 CP H ;ошибка?
460 JR NZ,QUIT ;если да, то закончить загрузку
470 PUSH HL ;сохранить HL
480 LD HL,(CNTLEN) ;уменьшить счетчик длины файла
490 DEC HL
500 LD (CNTLEN),HL
510 LD A,H ;проверить не достиг ли он 0
520 OR L
530 POP HL ;восстановить HL
540 EXX ;сменить набор регистров
550 JR Z,QUIT ;выход, если счетчик обнулится
560 POP AF ;восстановить AF
570 RET ;возврат из подпрограммы
;Закончить загрузку и выйти
580 QUIT LD SP,(MEM_SP) ;восстановили указатель стека
590 POP HL ;восстановили HL'
600 EXX ;поменяли регистры
610 LD A,H ;установка флагов
620 CP 1 ;правильности/неправильности считывания
630 RET ;возврат в точку вызова
;Процедура пользователя, "разбавленная" командами CALL LOAD1B - считывание
;одного байта.
;В распоряжении пользователя находятся регистры AF, BC, DE, HL, IY.
;(IX-1) указывает на предыдущий загруженный байт
;L' - текущий загруженный байт
;IX - адрес, куда запишется текущий байт командой CALL LOAD1B
;При выходе из этой процедуры загрузка файла продолжится как ни в чем не бывало.
640 MYPROC RET ;сейчас никакой процедуры пользователя нет
650 CNTLEN DEFW 0 ;счетчик длины загружаемого файла
660 MEM_SP DEFW 0 ;ячейка сохранения указателя стека
670 HEADER DEFS 17 ;буфер для загрузки заголовка
680 START EQU HEADER+13 ;адрес загрузки файла в заголовке
690 LENGTH EQU HEADER+11 ;длина файла в заголовке

```

Приводим дамп памяти приведенной программы:

7530	DD	21	A9	75	11	11	00	AF	92
7538	37	CD	56	05	30	F2	3A	A9	11
7540	75	FE	03	20	EB	DD	2A	B6	F3

7548	75	ED	5B	B4	75	3E	FF	37	17
7550	CD	59	75	CD	3F	05	D8	CF	18
7558	1A	F3	1B	ED	53	A5	75	11	60
7560	01	00	14	08	15	CD	62	05	3B
7568	3E	7F	DB	FE	1F	D0	7B	FE	DB
7570	01	D0	D9	E5	ED	73	A7	75	F0
7578	CD	A4	75	CD	80	75	18	FB	A8
7580	F5	D9	1E	01	CD	A9	05	BC	19
7588	20	10	E5	2A	A5	75	2B	22	A3
7590	A5	75	7C	B5	E1	D9	28	02	34
7598	F1	C9	ED	7B	A7	75	E1	D9	05
75A0	7C	FE	01	C9	C9	00	00	00	22
75A8	00	00	00	00	00	00	00	00	1D
75B0	00	00	00	00	00	00	00	00	25
75B8	00	00	00	00	00	00	00	00	2D

Описанная выше процедура имеет несколько точек входа:

1. Метка FILE - считывание первого попавшегося блока кодов с заголовком. При ошибке загрузки или нажатии BREAK (SPACE) выдаются соответствующие сообщения.

Вызывает процедуру ПЗУ по адресу 1366 (#0556) и затем переходит на метку LOADER.

2. Метка LOADER - загрузка, аналогичная CALL 2050 с выдачей сообщений об ошибке или нажатии BREAK.

Вход: IX - адрес загрузки блока кодов

DE - длина блока кодов

A=#00 - заголовок

A=#FF - файл

флаг C:включен - загрузка

выключен - проверка

3. Метка PROC - головная процедура загрузки. Ее вызывают все вышестоящие части программы. Входные данные аналогичны предыдущему пункту. Отличие состоит в том, что сообщения об ошибках не выдаются, хотя флаги и устанавливаются.

На выходе: флаг C включен - считывание данных произошло без ошибок;

флаг C выключен - произошла ошибка.

Во время запуска программы из любой точки входа выполняется следующее:

1. Считывается первый байт.
2. Вызывается процедура пользователя.
3. Если произошел возврат из пользовательской процедуры, то продолжается "обыкновенная" загрузка блока кодов до конца.
4. При появлении ошибки или нажатии BREAK производится экстренный выход из программы.

Процедура пользователя должна регулярно вызывать CALL LOAD1B.

Ограничения, накладываемые на процедуру пользователя:

1. В распоряжении пользователя регистры:
AF, BC, DE, HL - при выполнении CALL LOAD1B они не будут портиться;
B', E' - испортятся после CALL LOAD1B;
H' - контроль правильности загрузки - лучше не трогать;
L' - загруженный, но еще не записанный в память байт. Не менять!
D' - перед CALL LOAD1B всегда равен 0. Не менять!
C' - работа с бордюром. Не менять!
2. (IX-1) - указывает на предыдущий загруженный байт;
IX - указывает, куда будет записан только что загруженный байт;
3. IY - использовать, конечно, возможно, но лучше с BASIC'ом не конфликтовать;
4. SP - вообще лучше не трогать - плохо будет;
5. Во время работы программы, прерывания запрещены и разрешать их настоятельно не рекомендуется;
6. Вся процедура должна "разбавляться" командами CALL LOAD1B – чтение одного байта.

Для примера дальше приводится листинг процедуры пользователя. Для ее вызова необходимо заменить строку MYPROC RET на MYPROC JP PRINT

```

700 PRINT CALL    LOAD1B
710      LD      HL, (TXTADR)    ;адрес следующего символа
720      LD      A, (HL)        ;взяли символ
730      INC     HL             ;адрес следующего символа

```

```

740 CP 255 ;конец текста?
750 RET Z ;выход, если да
760 LD DE,PRINT ;заносим на стек адрес процедуры PRINT,
770 PUSH DE ;чтобы возвратиться по RET
780 CP 6 ;символ табуляции?
790 JR Z,SPCTAB ;переход
800 CP 22 ;смена позиции курсора?
810 JR Z,CH_POS
820 CP 16 ;смена атрибута печати?
830 JR Z,CH_ATR
840 CP 30 ;код паузы?
850 JR Z,PAUSE
860 CP 31 ;код вызова пользовательской процедуры?
870 JR Z,USERPR
880 LD (TXTADR),HL ;сохранили адрес текста
890 JR NC,PUTCHR ;печать, если код символа больше 31
900 LD HL,(C_COOR) ;считали координаты
910 CP 13 ;если перевод строки,
920 JR Z,ENTER ;то выполнить
930 CP 8 ;если шаг назад,
940 JR Z,BACK ;то выполнить
950 RET
960 SPCTAB LD C,(HL) ;процедура повторения печати одинаковых
;символов
970 INC HL ;в C - число символов
980 LD A,(HL) ;в A - код символа
990 PUSH HL
1000 LP1 PUSH AF
1010 CALL PUTCHR ;печать символа
1020 POP AF
1030 DEC C ;все распечатали?
1040 JR NZ,LP1 ;повторить, если не кончили
1050 POP HL
1060 JR GO1 ;переход на завершение
1070 PAUSE LD C,(HL) ;процедура паузы
1080 INC HL ;в BC - число считываемых байт
1090 LD B,(HL)
1100 INC HL
1110 LD (TXTADR),HL
1120 LPPAUS CALL LOAD1B
1130 DEC BC
1140 LD A,B
1150 OR C
1160 JR NZ,LPPAUS
1170 RET
1180 USERPR LD E,(HL) ;процедура вызова пользовательской
1190 INC HL ;функции
1200 LD D,(HL) ;в DE - адрес процедуры
1210 INC HL
1220 LD (TXTADR),HL
1230 PUSH DE ;занесли на стек адрес процедуры
1240 RET ;выход на процедуру
1250 CH_POS LD A,(HL) ;процедура смена позиции курсора
1260 AND 24 ;
1270 OR 64 ;
1280 LD D,A ;
1290 LD A,(HL) ; Расчет адреса верхней
1300 AND 7 ; линии знакоместа в новой
1310 RRCA ; позиции вывода
1320 RRCA ;
1330 RRCA ;
1340 INC HL ;
1350 OR (HL) ;
1360 LD E,A ;

```

```

1370     LD      (C_COOR),DE      ;сохранили адрес
1380 GO1  INC      HL
1390     LD      (TXTADR),HL
1400     RET
1410 CH_ATR LD      A,(HL)        ;смена атрибута печати
1420     LD      (ATTR),A        ;занести атрибут в переменную
1430     JR      GO1
1440 ENTER LD      A,L          ;процедура возврата каретки
1450     AND     224
1460     ADD     A,32
1470     LD      L,A
1480     JR      NC,GO3
1490     JR      GO2
1500 BACK LD      A,L          ;процедура возврата курсора на одну
1510     DEC     L              ;позицию назад
1520     OR      A
1530     JR      NZ,GO3
1540     LD      A,H
1550     SUB     8
1560     LD      H,A
1570     CP     64
1580     JR      NC,GO3
1590     JR      GO4
1600 PUTCHR LD     L,A          ;процедура печати символа
1610     LD      H,0
1620     ADD     HL,HL
1630     ADD     HL,HL
1640     ADD     HL,HL
1650     LD      DE,(CHARS)
1660     ADD     HL,DE
1670     LD      DE,(C_COOR)
1680     PUSH   DE
1690     LD      B,8
1700     CALL   LOAD1B
1710 LP2  LD      A,(HL)
1720     LD      (DE),A
1730     INC     HL
1740     INC     D
1750     DJNZ   LP2
1760     CALL   LOAD1B
1770     DEC     D
1780     LD      A,D
1790     RRCA
1800     RRCA
1810     RRCA
1820     AND     3
1830     OR     88
1840     LD      D,A
1850     LD      A,(ATTR)
1860     LD      (DE),A
1870     POP    HL
1880     INC     L
1890     JR      NZ,GO3
1900 GO2  LD      A,H
1910     ADD     A,8
1920     LD      H,A
1930     CP     88
1940     JR      C,GO3
1950 GO4  LD      H,64
1960 GO3  LD      (C_COOR),HL
1970     RET
1980 TXTADR DEFW   TEXT        ;адрес следующего символа
1990 C_COOR DEFW   #4000      ;адрес вывода на экран
2000 ATTR DEFB    7          ;атрибуты печати

```

```

2010 CHARS DEFW 15360 ;адрес символьного набора
2020 USER_1 LD C, (HL) ;пример пользовательской процедуры
2030 INC HL
2040 LD B, (HL)
2050 INC HL
2060 LD (TXTADR), HL
2070 LPUSR1 CALL LOAD1B
2080 LD A, R
2090 OUT (254), A
2100 DEC BC
2110 LD A, B
2120 OR C
2130 JR NZ, LPUSR1
2140 RET
2150 TEXT DEFB 30
2160 DEFW 2048
2170 DEFB 22, 0, 0, 16, 7
2180 DEFB 6, 32, "-"
2190 DEFM "HELLO, ANYBODY!"
2200 DEFB 13, 6, 32, "="
2210 DEFB 31
2220 DEFW USER_1, 500
2230 DEFB 13, 16, #38
2240 DEFM "IT'S ME - SERZHSOFT"
2250 DEFB 31
2260 DEFW USER_1, 400
2270 DEFB 8, 8, 8, 8
2280 DEFM "COMPANY"
2290 DEFB 255

```

Итак, в тексте могут встречаться следующие управляющие коды:

#FF	конец текста
#10 #NN	новый атрибут печати #NN
#1E #LOW #HIGH	считать (LOW+256*HIGH) байт
#1F #LOW #HIGH	вызвать процедуру пользователя по адресу(LOW+256*HIGH)
#06 #NN #MM	печать #NN раз символ с кодом #MM
#16 #YY #XX	переход к координатам печати #XX, #YY
#0D	возврат каретки + перевод строки
#08	возврат на одну позицию назад

Дальше идет дамп процедуры вместе с процедурой загрузки:

```

7530 DD 21 AB 75 11 11 00 AF 94
7538 37 CD 56 05 30 F2 3A AB 13
7540 75 FE 03 20 EB DD 2A B8 F5
7548 75 ED 5B B6 75 3E FF 37 19
7550 CD 59 75 CD 3F 05 D8 CF 18
7558 1A F3 1B ED 53 A7 75 11 62
7560 01 00 14 08 15 CD 62 05 3B
7568 3E 7F DB FE 1F D0 7B FE DB
7570 01 D0 D9 E5 ED 73 A9 75 F2
7578 CD A4 75 CD 80 75 18 FB A8
7580 F5 D9 1E 01 CD A9 05 BC 19
7588 20 10 E5 2A A7 75 2B 22 A5
7590 A7 75 7C B5 E1 D9 28 02 36
7598 F1 C9 ED 7B A9 75 E1 D9 07
75A0 7C FE 01 C9 C3 BC 75 00 4D
75A8 00 00 00 00 00 00 00 00 1D
75B0 00 00 00 00 00 00 00 00 25
75B8 00 00 00 00 00 CD 80 75 2A 19
75C0 8D 76 7E 23 FE FF C8 11 AF
75C8 BC 75 D5 FE 06 28 21 FE 8E
75D0 16 28 45 FE 10 28 59 FE 55
75D8 1E 28 24 FE 1F 28 30 22 4E
75E0 8D 76 30 6B 2A 8F 76 FE 20
75E8 0D 28 4B FE 08 28 51 C9 25

```

75F0	4E	23	7E	E5	F5	CD	4F	76	C0
75F8	F1	0D	20	F8	E1	18	2C	4E	F6
7600	23	46	23	22	8D	76	CD	80	74
7608	75	0B	78	B1	20	F8	C9	5E	66
7610	23	56	23	22	8D	76	D5	C9	E5
7618	7E	E6	18	F6	40	57	7E	E6	FB
7620	07	0F	0F	0F	23	B6	5F	ED	EF
7628	53	8F	76	23	22	8D	76	C9	07
7630	7E	32	91	76	18	F5	7D	E6	CD
7638	E0	C6	20	6F	30	4B	18	3F	B5
7640	7D	2D	B7	20	44	7C	D6	08	D5
7648	67	FE	40	30	3C	18	38	6F	8E
7650	26	00	29	29	29	ED	5B	92	41
7658	76	19	ED	5B	8F	76	D5	06	85
7660	08	CD	80	75	7E	12	23	14	67
7668	10	FA	CD	80	75	15	7A	0F	48
7670	0F	0F	E6	03	F6	58	57	3A	CC
7678	91	76	12	E1	2C	20	0A	7C	BA
7680	C6	08	67	FE	58	38	02	26	E1
7688	40	22	8F	76	C9	A8	76	00	4C
7690	40	07	00	3C	4E	23	46	23	63
7698	22	8D	76	CD	80	75	ED	5F	41
76A0	D3	FE	0B	78	B1	20	F4	C9	F8
76A8	1E	00	08	16	00	00	10	07	71
76B0	06	20	2D	48	45	4C	4C	4F	ED
76B8	2C	20	41	4E	59	42	4F	44	37
76C0	59	21	0D	06	20	3D	1F	94	D3
76C8	76	F4	01	0D	10	38	49	54	9B
76D0	27	53	20	4D	45	20	2D	20	DF
76D8	53	45	52	5A	48	53	4F	46	C2
76E0	54	1F	94	76	90	01	08	08	74
76E8	08	08	43	4F	4D	50	41	4E	2C
76F0	59	FF	00	00	00	00	00	00	BE

Теперь можно неограниченно расширять процедуру, но лучше загружать ее в меньшие адреса, т.к. там программы работают быстрее, хотя не на всех компьютерах. Также, можно применять чаще процедуру загрузки одного байта, если происходит сбой, и программа теряет считываемый байт. Реализуйте возможности, которые нужны, но не используйте процедуры, использующие прерывания.

© Матюшев Александр, г. Сланцы, 1996.

ПРОЦЕДУРА ПЕЧАТИ 42 СИМВОЛОВ В СТРОКЕ.

В ZX-РЕВЮ 95/5 на стр. 29 Вы поместили статью Сергеева Дмитрия "Процедура печати 42 символов в строке". Идея, заложенная в процедуре, мне понравилась. Давно хотелось иметь для использования в своих программах быстродействующую процедуру печати 42 символов в строке, полностью независимую от ПЗУ компьютера. Никак не хочу обидеть автора программы, но детально разобрав работу процедуры, удалось путем замены некоторых команд, новой процедуры расчета адреса атрибутов (позаимствована у Патрахина В.В., ZX-РЕВЮ 94/6, стр. 34), использования управляющих кодов в тексте, размещения переменных "ST" и "MNB" прямо в процедуре, существенно сократить объем и увеличить быстродействие процедуры. Использование управляющего кода "AT" (#16) осталось таким же, управляющие коды "INK", "PAPER", "BRIGHT", "FLUSH" объединены и рассчитываются стандартно: INK+PAPER*8+BRIGHT*64+FLUSH*128. В тексте это число должно стоять после кода 16(#10). Встретив его, процедура положит следующее за ним число в переменную ATTR_T (23695=#5C8F) и установит временные атрибуты печати. В итоге получилось следующее:

```
0010   ORG      50000
0020   ENT
0030   LD      HL,TEXT
0040 WRITE LD   A,(XTXT)
0050   CP      42
0060   JR      C,WR2
0070   LD      A,(YTXT)
0080   INC     A
0090   CP      24
0100   JR      C,WR1
```

```

0110      XOR      A
0120 WR1  LD      (YTXT),A
0130      XOR      A
0140      LD      (XTXT),A
0150 WR2  LD      A,(HL)
0160      INC     HL
0170      AND     A
0180      RET     Z
0190      CP      32
0200      JR      C,WRIT2
0210 WRIT1 CALL  PRSYM
0220      JR      WRITE
0230 WRIT2 CP      16
0240      JR      NZ,WRIT3
0250      LD      A,(HL)
0260      LD      (23695),A
0270      INC     HL
0280      JR      WR2
0290 WRIT3 CP      22
0300      JR      NZ,WRIT4
0310      LD      A,(HL)
0320      LD      (YTXT),A
0330      INC     HL
0340      LD      A,(HL)
0350      LD      (XTXT),A
0360      INC     HL
0370      JR      WRITE
0380 WRIT4 LD      A,"?"
0390      JR      WRIT1
0400 PRSYM EXX
0410      PUSH    HL
0420      LD      L,A
0430      LD      H,0
0440      ADD     HL,HL
0450      ADD     HL,HL
0460      ADD     HL,HL
0470      LD      DE,15360      ;FONT-256
0480      ADD     HL,DE
0490      PUSH    HL
0500 CHECK LD      DE,(XTXT)
0510      LD      A,E
0520      ADD     A,A
0530      LD      E,A
0540      ADD     A,A
0550      ADD     A,E
0560      LD      E,255
0570      LD      L,8
0580 CHPL1 INC     E
0590      SUB     L
0600      JR      NC,CHPL1
0610      ADD     A,L
0620      LD      H,A
0630      LD      A,L
0640      SUB     H
0650      LD      (SL1-1),A
0660      LD      (SL2-1),A
0670 ADDRZ LD      A,D
0680      AND     #18
0690      ADD     A,#40
0700      LD      H,A
0710      LD      A,D
0720      AND     7
0730      RRCA
0740      RRCA

```

```

0750      RRCA
0760      ADD      A, E
0770      LD        L, A
0780      PUSH     HL
0790 AADDR LD      A, H
0800      RRCA
0810      RRCA
0820      RRCA
0830      AND      #03
0840      OR       #58
9850      LD        H, A
0860 ATTR LD      A, (23695)
0870      LD        (HL), A
0880      POP      HL
0890      POP      DE
0900      LD        C, 8
0910 SCRL LD      A, (DE)
0920      LD        B, 0
0930 SL1 SRL      (HL)
0940      DJNZ     SL1
0950      AND      A
0960      LD        B, 0
0970 SL2 RL       A
0980      RL      (HL)
0990      DJNZ     SL2
1000      INC      HL
1010      LD        (HL), A
1020      DEC      HL
1030      INC      H
1040      INC      DE
1050      DEC      C
1060      JR       NZ, SCRL
1070 FINIS LD      HL, XTXT
1080      INC      (HL)
1090      POP      HL
1100      EXX
1110      RET
1120 XTXT DEFB    0
1130 YTXT DEFB    0
1140 TEXT DEFS    0
START OF OBJECT: #C350 (50000)
LENGS          : #00AF (00175)

```

Если использовать символьный набор шириной 6 пикселей, прижатый к левому верхнему углу знака, можно изменить строку N900 на LD C,6, что сократит время выполнения цикла. Если не предусматривается возврат в BASIC, строки 410 и 1090 можно убрать, если нет необходимости сохранения регистров, которые будут испорчены процедурой, можно убрать строки 400 и 1100.

© Alex, г. Нижний Тагил, 1996.

НАСТРОЙКА ДИСКОВОДОВ.

Я занимаюсь настройкой дисководов уже несколько лет, и разработал целую систему настройки шагового двигателя, да и самого дисковода.

Предлагаю Вашему вниманию методику настройки 11-х и 13-х дисководов. Начнем с 13-х дисководов.

У многих USER'ов возникла мощная проблема: HERO QUEST и ELITE не хотят отгружаться на диск. В SPECTROFON'e посоветовали форматировать диски разнообразными мудренными способами. И в том же SPECTROFON'e проскользнуло упоминание о некой загадочной перемычке, которую нужно пере-ставить в другое положение.



Перемычка должна находиться в положении "ON", тогда все программы будут нормально отгружаться на диск и улучшится качество чтения.

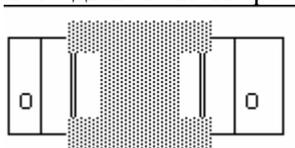
ИФК: В некоторых случаях более эффективным оказывается полное удаление данной перемычки с платы.

КОРР: Ну, с перемычками разобрались, теперь о настройке шагового двигателя:

Шаговый двигатель настроить не очень сложно, особенно на 11-х и 13-х дисководах, исключение составляют 5-е дисководы.

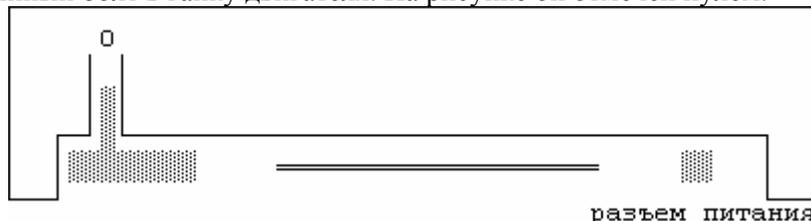
Система настройки:

1) Нужно ослабить крепежные болты шагового двигателя. На рисунке они отмечены нулями.



2) Сдвинуть двигатель до упора по направлению к замку дисковода.

3) Далее вкрутить длинный болт в гайку двигателя. На рисунке он отмечен нулем.



4) Вращая этот болт, добиться отличного качества чтения. Для проверки чтения желательно использовать программу DCU 8.02.

5) Потом закрутить крепежные болты потуже.

Для настройки дисковода желательно иметь и второй дисковод, с которого, в случае необходимости, можно подгрузить DCU.

Если Вы подключили второй дисковод, то из одного из них следует вытащить блок резисторов, который показан на первом рисунке.

Теперь о настройке 11-х дисководов.

11-й дисковод настраивается по тому же принципу, что и 13-й, но он еще имеет электромагнит, который опускает головки. У многих моих знакомых, купивших 11-й дисковод, не опускалась головка на диск. Для устранения этой неисправности нужно поставить перемычку. Она показана на первом рисунке.

© Комиссаров Сергей, г. Дмитровград.

Хочу внести свой небольшой вклад по программе G.A.C. и посоветовать Олегу Бобрышову из г. Георгиевска (ZX-PEBЮ 96/1-2, стр.102).

Дело в том, что у Олега, как я понял, программа G.A.C. из журнала "SPECTROFON", и если ее запустить не из TR-DOS, а из бейсика (RANDOMIZE USR 15619 : REM : RUN "G.A.C.D."), то в режиме сообщений не возникнет никаких проблем. Можно набирать текст в семь строк, как и написано в инструкции.

© Лебедев Павел, 1996

Недавно мне понадобилась как можно более подробная информация по программированию КР1818ВГ93 и я полез в сборники ZX PEBЮ... Вот тут-то все и началось!

Сначала я хочу сказать о вызове подпрограмм TR-DOS. Время от времени в различных книгах и программах вместо

```
PUSH    HL
JP      #3D2F
```

встречается

```
PUSH    HL
JP      #3D30
```

Хочу предостеречь людей от таких "улучшений". Несмотря на то, что по адресу #3D2F в TR-DOS находится команда NOP, а RET - в следующей ячейке, передавать управление необходимо именно на адрес #3D2F. Дело в том, что в некоторых компьютерах ПЗУ не успевают переключиться достаточно быстро, и процессор как бы "проскакивает" команду RET. Последствия этого могут быть самыми печальными.

Теперь несколько слов об управлении KP181BG93. Практически во всех изданиях, посвященных этому вопросу, сказано, что напрямую прочитать регистр состояния из порта #1F невозможно. Каково же было мое удивление, когда поковырявшись в своей TR-DOS 5.03 по адресу #09BF я обнаружил следующие команды:

```
09BF    IN      A, (#1F)
09A1    RET
```

Возможно, это имеет место только в TR-DOS 5.03.? В связи с этим, а также в целях создания более качественного программного обеспечения для работы с KP181BG93 программистами СНГ, предлагаю открыть раздел, в который все желающие могли бы прислать адреса необходимых подпрограмм для своей версии ПЗУ, чтобы их версия TR-DOS поддерживались новыми программами.

Предлагаю точки входа для TR-DOS 5.03:

1. Вывод в порт BC регистра A:

```
2A53    OUT     (C), A
2A55    RET
```

2. Вывод в порт #1F (регистр команд) регистра A:

```
2FC3    OUT     (#1F), A
2FC5    RET
```

3. Вывод в порт #FF регистра A:

```
1FF3    OUT     (#FF), A
1FF5    RET
```

4. Чтение регистра состояния (порт #1F) в регистр A:

```
09BF    IN      A, (#1F)
09A1    RET
```

Теперь несколько слов о самом загадочном Спектруме - НОВВИТ'е. В дополнение к сказанному Наилем Исяновым в ZX РЕВЮ 96/1-2, хочу добавить, что все прошивки ПЗУ в нем более, чем нестандартные. В бейсик-редактор встроено несколько дополнительных возможностей вроде вызова последней введенной команды и т.д. Плюс поддержка русского языка (добавлены контрольные коды #1E - РУС и #1F - ЛАТ). По LOGO-RUS мне известны следующие команды: ВПn - вперед на n точек, НЗn - назад на n точек, Пpн - поворот вправо на n градусов, ЛВn - поворот влево на n градусов, СТР - включить режим стирания.

Создание процедур осуществляется следующим образом:

```
ЭТО    <имя>
.....
КОНЕЦ
```

где <имя> - имя процедуры. После ввода такой последовательности на экран выводится сообщение "ТЕПЕРЬ ПОНЯТНО ЧТО ТАКОЕ <имя>", и Вы можете в качестве команды использовать <имя>.

А теперь самое интересное: о TR-DOS НОВВИТ'а. На Technology Research там нет ни малейшего намека, по крайней мере в незашифрованном виде, зато по адресу #3BC6 обнаружено следующее: "Complex 1991". А по адресу #3BE5 расположена и вовсе шокирующая надпись: "Written by D.I.Mikhailov, Saint-Petersburg, USSR. Tel (812) 554-26-13 GOOD LUCK !". Кроме того, по всей ПЗУ раскиданы обращения к портам #F6 и #F7, а иногда встречается и такое:

```
097F    IN      A, (#1F)
0981    AND     C
0982    LD      (#E5EB), A
0985    RET     Z
0986    LD      A, (#E5EA)
0989    DEC     A
098A    LD      (#E5EA), A
```

Обратите внимание на адреса данных!!! Я, конечно, понимаю, что если 64К ОЗУ, то все можно, но не до такой же степени!

И наконец - хочу привести усовершенствованную версию Advanced Drive FX, опубликованного в ZX РЕВЮ 95/5. Новая версия несколько более понятна, меньше по объему, позволяет оперировать со всеми дисководами от А до D и изменять скорость зажигания и затухания светодиода. Для зажигания необходимо вызвать

процедуру F_IN, а для затухания - F_OUT. Подпрограммы работы с портами настроены на TR-DOS 5.03. Для работы с другими версиями надо изменить адреса в строках 390-410. В строке 420 регулируется скорость эффекта, а в строке 430 задается дисковод.

```

10      ORG      50000
20 F_IN  LD      E,1          ;зажигание производится при изменении переменной
                                ;на +1
30      JR      DO_FX        ;переход на процедуру
40 F_OUT LD      E,255       ;затухание производится при изменении
                                ;переменной на -1
50 DO_FX XOR     A           ;подготовка аккумулятора
60 LOOP1 ADD    A,E         ;изменяем переменную
70      RET     Z           ;выход, если светодиод зажегся или потух
                                ;полностью
80      LD      B,SPEED      ;взяли скорость загорания (затухания)
90 LOOP2 PUSH   BC         ;сохранили счетчик
100     LD      D,A         ;сохранили указатель загорания (затухания)
110     LD      HL,OUT_FF    ;адрес подпрограммы вывода регистра A в
                                ;системный регистр (порт #FF)
120     LD      A,#3C       ;в аккумуляторе необходимое число, которое
                                ;указывает на номер дисковода: биты 0 и 1.
130     CALL   DOS          ;заносим число в порт
140     LD      HL,OUT_1F    ;выдаем команду контроллеру "Чтение
                                ;адреса"
150     LD      A,#D0
160     CALL   DOS
170 WAIT LD      HL,IN_1F    ;чтение регистра состояния
180     CALL   DOS
190     XOR     #80         ;проверка готовности дисковода
200     JR      NZ,WAIT     ;если не готов, то ждать
210     LD      A,8         ;выдаем команду "Восстановление"
220     LD      HL,OUT_1F
230     CALL   DOS
240     LD      B,D         ;задержка
250     DJNZ   $
260     LD      HL,OUT_FF    ;выбираем нужный дисковод
270     LD      A,DRVNUM
280     CALL   DOS
290     LD      A,D         ;восстановили указатель
300     NEG    ;изменили знак
310     LD      B,A         ;задержка
320     DJNZ   $
330     LD      A,D         ;восстановили указатель
340     POP   BC           ;восстановили счетчик
350     DJNZ   LOOP2       ;повтор, пока счетчик не обнулится
360     JR      LOOP1      ;переход на начало процедуры
370 DOS  PUSH   HL         ;процедура вызова подпрограмм из ПЗУ TR-DOS
380     JP     #3D2F
390 IN_1F EQU    #09BF     ;адрес подпрограммы чтения системного регистра
                                ;необходимо изменить под конкретный тип ПЗУ
400 OUT_1F EQU   #2FC3     ;выдача команды для контроллера
410 OUT_FF EQU   #1FF3     ;запись системного регистра
420 SPEED EQU    #08       ;скорость эффектов
430 DRVNUM EQU   #00       ;номер дисковода

```

Ниже приводится дамп программы:

C350	1E	01	18	02	1E	FF	AF	83	9B
C358	C8	06	08	C5	57	21	F3	1F	40
C360	3E	3C	CD	96	C3	21	C3	2F	D6
C368	3E	D0	CD	96	C3	21	BF	09	48
C370	CD	96	C3	EE	80	20	F6	3E	1B
C378	08	21	C3	2F	CD	96	C3	42	BE
C380	10	FE	21	F3	1F	3E	00	CD	8F
C388	96	C3	7A	ED	44	47	10	FE	A4
C390	7A	C1	10	C7	18	C1	E5	C3	E6

Не забудьте, что при использовании других версий TR-DOS надо изменить адрес в строке 390. Для того чтобы узнать, есть ли у Вас в ПЗУ такая подпрограмма, необходимо сначала выгрузить ПЗУ TR-DOS на диск. Для этого войдите в TR-DOS и наберите команду SAVE "TRDOS"CODE 0,16384.

Затем загрузите этот блок в память и наберите небольшую программку:

```
5 RANDOMIZE USR 15619: REM : LOAD "TRDOS" CODE 49152
10 FOR N=49152 TO 65535
20 IF PEEK N=219 AND PEEK (N+1)=31 AND PEEK (N+2)=201 THEN PRINT AT 10,10;N-
49152:PAUSE 0
30 NEXT N
```

В процессе работы программы на экран может быть выведен адрес, который необходимо подставить в строку 390. Если адрес не будет выведен, то нужно будет использовать другой доступ к системному регистру, например, как в предыдущей версии программы, которая была опубликована в ZX РЕВЮ 95/5.

ИФК: Андрей Рубин из г. Энгельса прислал дополнительную информацию по недокументированным операторам LASER BASIC-а.

КОРР: Функции .RLCT и .MLN неразрывно связаны, т.к. .RLCT - оператор, который осуществляет перемещение спрайт-файла в памяти, а в переменной .MLN задается смещение в байтах (.MLN=[смещение в байтах], если смещение идет вверх, и .MLN=65536-[смещение в байтах], если вниз). Вот почему нет "обратной связи" LET<переменная Бейсика>=?MLN.

ИФК: потребность в таком перемещении может возникнуть, например, при необходимости зарезервировать область памяти между верхней границей области спрайтов и началом интерпретатора (например, для каких-нибудь дополнительных подпрограмм).

ИФК: Андрей Рубин задает, также, очередную задачку любителям поломать голову.

КОРР: Загрузите любую программу на BASIC-е, остановите ее и войдите в TR-DOS (впрочем, можно просто набрать строку: 10 REM I AM IN TR-DOS !!!). Нажмите [EDIT] - эта строка (или любая, на которой стоит курсор) появится в командной строке DOS. Почему так происходит? Ведь SOS и не пахнет! К тому же, если Вы строку измените, а потом нажмете [ENTER], то в памяти строка останется неизменной - т.е., фактически после нажатия [ENTER] просто происходит очистка командной строки. Так что же все-таки происходит?

ИФК: Кроме того, используя клавиши "курсор вверх" и "курсор вниз" можно просмотреть всю BASIC программу.

ИФК: Интересное письмо с информацией по "железу" пришло к нам от В.А. Кекуха из г. ТУЛУН.

КОРР: я хочу предложить информацию, которая, возможно, заинтересует пользователей ZX-SPECTRUM с дисковым интерфейсом. У меня дисковод "Электроника МС 5313", и я стал замечать за ним интересные явления. Он начал портить системный трек, в особенности, девятый сектор нулевой дорожки.

ИФК: Напомним, что девятый (служебный) сектор используется системой для хранения информации о самой дискете.

КОРР: я, кажется, нашел этому объяснение. В дисковом есть оптопара, состоящая из светодиода и фотодиода, а у головок есть движок, который заходит между элементами оптопары, когда головка попадает на нулевую дорожку, и в этот момент в контроллер идет сигнал TRACK ZERO. Так вот, стоит винтам крепления ослабнуть, как оптопара под действием пружинок отодвигается, и головки могут вообще не попасть на нулевую дорожку.

ИФК: Происходит ошибка позиционирования, хотя контроллер и "считает", что правильно установил головку дисковода.

КОРР: В этом случае нужно отвернуть 2 винта крепления, подвинуть ее поближе к корпусу и закрепить в этом положении винтами. Возможно, я ошибаюсь, но из 5 человек, которых я знаю, у двоих была такая неисправность, и после регулировки все стало нормально.

ИФК: письмо на актуальную тему пришло от Ильи Пивена (Selina Space Studio).

КОРР: разбирая работу, изложенных в "ЭТЮДАХ" программ, связанных с увеличенными символами: печать символов двойной высоты и пр., я наткнулся на один интересный факт, о котором в выпусках РЕВЮ совсем нет информации. Речь идет вот о чем. Ни одна из оттестированных программ не учитывала того, что экранная область памяти SPECCY сегментирована. Это не имеет значения тогда, когда печать выполняется внутри сегментов, но никак не на их границе.

ИФК: Подробно вопросы сегментации экрана рассмотрены в книге "Элементарная графика".

КОРР: таким образом, есть две "опасные" Y-координаты - 7 и 15, и, соответственно, две возможности некорректной работы программы печати. Для наиболее простой проверки вышеизложенного, рекомендую в программе печати Игоря Комарова (ZX РЕВЮ 95/2, стр.45) в строке 230 подставить "опасные" Y-координаты. Наиболее быстрый и доступный способ решения проблемы такой экранной несовместимости - специально указывать, что какая-либо программа не учитывает сегментированности экранной памяти SPECCY. Но это никоим образом не решает проблему в корне. Решить ее - значит создать дополнительные процедуры проверки Y-координаты позиции печати и соответствующей настройки программы.

ИФК: действительно, существуют два пути решения: либо вставлять в программу дополнительные операторы проверки, либо указывать на ограничения в применении данной программы. Все зависит от задач, которые Вы ставите перед собой при написании программы. Если Вы четко знаете, что печать текста будет происходить только в одном сегменте, то встраивать дополнительные операторы проверки - только лишний расход памяти. Если же Вы собираетесь написать универсальную программу с возможностью печати в любой точке экрана, тогда Вам волей-неволей придется решать проблемы, связанные с сегментированием экрана.

ИФК: письмо на тему: "магнитофонный загрузчик с индикатором загрузки" пришло к нам от Артемьева Богдана из г. Жердевка.

КОРР: мне часто встречаются игры без рисунка при загрузке и, согласитесь, надоедает сидеть перед таким экраном и гадать: скоро ли конец загрузки. В одно из таких ожиданий я и задумал программу, которую и предлагаю сейчас вашему вниманию. Это ленточный индикатор загрузки. Он загружает блок кодов без заголовка, и по мере загрузки "заполняет" шкалу на экране. Перед вызовом программы необходимо ввести длину (LONG) и адрес загрузки блока (NACH.). При желании, можно добавить в начале программы блок загрузки заголовка и расчет LONG и NACH. относительно него.

```
LD      (IY+83), 6
LD      (IY+14), 6
LD      A, 2
CALL    5633
CALL    3435
LD      HL, LONG+1
INC     (HL)
LD      BC, 12343
CALL    8933
LD      B, 0
LD      A, (LONG+1)
LD      C, A
PUSH    BC
LD      DE, 65281
CALL    9402
LD      BC, 512
PUSH    BC
CALL    9402
LD      E, D
PUSH    DE
LD      A, (LONG+1)
INC     A
INC     A
LD      C, A
LD      B, 0
PUSH    BC
CALL    9402
LD      BC, 2816
LD      DE, 257
CALL    9402
POP     BC
CALL    9402
POP     DE
POP     BC
CALL    9402
POP     BC
CALL    9402
LD      BC, 1792
```

```

CALL    9402
EXX
LD      DE,18919      ; адрес левого верхнего угла шкалы.
LD      A, (LONG+1)
LD      C,A          ; длина шкалы.
EXX
LD      IX, (NACH.)
LD      DE, (LONG)
XOR     A
LD      D,A
SUB     1
EX      AF,AF'
DI
CALL    #0562
LOOP    EXX
        PUSH    DE
        LD      A, (DE)
        RRCA
        SET     7,A
        LD      B,6
L1      LD      (DE),A
        INC     D
        DJNZ   L1
        POP     DE
        BIT    0,A
        JR     Z,P1
        INC     DE
P1      DEC     C
        JR     Z,P2
        EXX
        INC     D
        CALL   #05A9
        JR     LOOP
P2      LD      HL,10072
        EXX
        EI
        RET
LONG    DEFW           ; длина блока.
NACH    DEFW           ; начальный адрес загрузки.

```

ИФК: Мы протестировали эту программу, и она нам понравилась. По нашему мнению такого типа загрузчиками желательно оснащать все магнитофонные версии программ.

© Александр Андрей, г. Липецк, 1996.

Откликаюсь на просьбу о помощи читателей из п. Белогорск, Кемеровской обл. (ZX-РЕВЮ 96/1-2, с. 94-96). Тоже пытаюсь запрограммировать игру "LIFE", и могу посоветовать следующее.

Главная их ошибка в том, что они обсчитывают "вакуум", из за чего и возникают проблемы с быстродействием и движущимися фигурами (глайдерами). Вкратце, алгоритм моей программы такой. Чтобы не обсчитывать "вакуум", я отвел под координаты точек область памяти объемом 5К, т.е. 2,5К отводится под координату X и 2,5К отводится под координату Y. Таким образом, здесь умещаются координаты 2500 точек. Этого вполне хватает. Сама программа состоит из следующих блоков:

1) Определение координат соседних точек. Для каждой точки (x,y) таких точек 8, т.е. (x-1,y), (x+1,y), (x,y-1), (x,y+1), (x-1,y-1), (x-1,y+1), (x+1,y-1), (x+1,y+1). Для этого массива координат отводится 20К памяти. Здесь сразу исключаются координаты, в которых уже есть точки фигуры, т.е. остаются только координаты соседних пустых клеток.

2) Происходит компрессия полученного массива соседних пустых клеток, т.к. более половины из полученных координат повторяются, и не один раз (ведь точки исходной фигуры находятся по соседству).

В ходе дальнейшей работы над "LIFE" блоки 1 и 2 я объединил. Это сократило объем программы и повысило быстродействие. Теперь вычисленная координата соседней точки сразу сравнивается с координатами точек фигуры и с ранее вычисленными координатами соседних точек и, в случае совпадения, исключаются.

3) Определение, в каких из пустых клеток образуются новые точки.

4) Определение, какие из старых точек выживают.

5) Объединение точек, полученных в блоках 3 и 4, в результате чего образуется искомая фигура (делается следующий ход).

В своей программе я широко применял команду блочного сравнения SPIR. Благодаря разделению в памяти координат X и Y, сравниваются сначала координаты X, а затем, при необходимости, координаты Y. Такая организация памяти удобна для загрузки и выгрузки фигур через магнитофон, т.к. не надо оперировать со всем экраном, а понадобится лишь количество точек, номер хода и координаты точек.

Что касается глайдеров, то их можно либо удалять "вручную", записывая в соответствующие им координаты нули, либо предусмотреть уничтожение точек возле границ поля по команде.

Диденко Алексей. 1996 г.

Решил написать вам, когда прочитал статью "ПРОБЛЕМЫ РАЗВИТИЯ ZX SPECTRUM" в ZX-РЕВЮ 96/1-2. Хочу высказать свою точку зрения на этот счет.

Во-первых, на Спрессу действительно неудовлетворительная ситуация с языками программирования, и решать эту проблему надо уже сейчас. Во-вторых, на просторах нашей Родины наблюдается такое разнообразие всевозможных моделей компьютеров и периферии к ним, что SPECTRUM практически исчез - остались только SPECTRUM-совместимые. Стандарта нет, и это, к сожалению, объективная реальность. Многие модели считаются перспективными, но практически везде имеются досадные недоработки. И хотя на страницах ZX РЕВЮ не раз обсуждались пути их устранения, я считаю, что подобное положение вещей недопустимо для любого коммерческого продукта. Не предлагает же фирма IBM взять пользователю в руки паяльник и исправить пару ошибок в схеме PENTIUMа.

Теперь перехожу от проблем к возможным путям их решения.

1. Необходимо существенно обновить зашитые в ПЗУ BASIC и DOS. Автор статьи в ZX-РЕВЮ предлагает в качестве ZX-SHELL использовать IS-DOS, но я с ним не согласен, т.к., по моему мнению, при разработке IS-DOS, не смотря на все ее достоинства, было допущено несколько просчетов:

а) в этой системе совмещены DOS и файловая оболочка, что крайне затрудняет написание альтернативных оболочек под IS-DOS, а существующая имеет ряд ограничений: невозможно копировать каталоги по клавише '5', невозможно удалить каталог с файлами, невозможно удалить файл, помеченный 'Read only' (нужно обязательно изменить его атрибуты).

б) имеется ряд атавизмов MS-DOS, например, дата создания файла, совершенно бессмысленные на Спрессу. Можно, конечно, возразить, что в некоторых моделях есть энергонезависимая память, но IS-DOS создавалась в те времена, когда этих компьютеров еще не было, да их и сейчас мало.

в) IS-DOS больше совместима с MS-DOS, а не с TR-DOS, что создает трудности при адаптации программ.

Поэтому мне кажется, что нужно разработать новую систему на основе TR-DOS, которая сможет работать с каталогами и, возможно, сегментированными файлами.

2. В последнее время появилось несколько интересных разработок для SPECTRUM'а. Прежде всего, хочется отметить гениальную идею С. Веремеенко по адаптации к Спрессу видеопроцессора. Я считаю, что подобные разработки надо всячески поддерживать. Но мне кажется, что видеопроцессор не станет стандартом, если дело ограничится опубликованием схемы. Я считаю, что необходим проект по разработке нового SPECTRUM-совместимого компьютера со следующими характеристиками:

- Процессор Z80H 7 МГц.
- ОЗУ 512Кб-1Мб.
- Видеопроцессор.
- Теневое ПЗУ с набором ПО.
- Альтернативное ПЗУ OS и DOS.
- Винчестер.
- И, возможно, ZX-COVOX.

Хочу добавить, что я готов участвовать в обсуждении и реализации подобного проекта.

ИГРЫ

ИФК: Наш старый знакомый, Илья Командин (ZX-MANIACS) из г. Калуга описывает игру THE SCEPTRE.

Корр: Действие игры происходит, судя по косвенным данным, на некоем острове, где обосновались семеро лордов со своими "армиями". Это UMBARG, VERDAN, TYRIAN, KUANO, AURIC, GRISUL и Вы

сами. Вашим прямым врагом является UM- BARG (его воины - самые темные), остальные же лорды нейтральны, хотя их воины тоже недолюбливают Вас и частенько нападают на Ваших людей). У каждого лорда имеется 8 типов воинов (расположены в порядке убывания силы):

THANE - тан, вождь;
ASSASSIN - убийца;
REAPER - жнец;
FOOL - безумец;
SAVAGE - дикарь;
MYSTIC - мистик, колдун;
HERALD - геральд;
THRALL - раб.

Каждый воин обладает определенными качествами. От них зависит его умение сражаться, скорость ходьбы, умение разговаривать с другими. Всего существует 22 качества:

LOYAL - преданный;
UNRULY - непокорный;
MERCIFUL - милосердный;
INCISIVE - пронизательный;
RUTHLESS - безжалостный;
BOLD - дерзкий;
BLUNT - грубый;
FOOLISH - глупый;
MASTERFUL - властный;
PERSUASIVE - убедительный;
FEEBLE - слабый;
OBEDIENT - послушный;
TIMID - робкий;
TREACHEROUS - предательский;
BRAVE - храбрый;
SWIFT - быстрый;
STRONG - сильный;
TIRELESS - неутомимый;
WISE - мудрый;
FLAGGIN - великолепный;
COWARDLY - трусливый;
SLOW - медлительный.

Возможно, существуют и другие качества, с которыми мы просто не сталкивались. От этих характеристик очень многое зависит. Так, например, у раба с качествами BRAVE и STRONG есть неплохой шанс уничтожить вражеского геральда, дикаря, а, подчас, и безумца.

Управлять своими воинами напрямую Вы не можете, поэтому общение с программой происходит с помощью приказов, которые Вы можете отдавать своим воинам. Приказов уйма, и с большинством из них мы не разобрались. Все приказы можно условно разбить на три группы. Первая группа - приказы, направленные на других воинов. Это:

FOLLOW - преследовать;
PROTECT - защищать;
STALK - подкрадываться;
HARASS - беспокоить;
AVOID - избегать;
FIND - найти;
KILL - убить;
BEWITCH - заколдовать;
PETRIFY - превратить в камень;
CURSE - проклясть;
CHARM - наслать чары;
RELEASE - освободить;
THREATEN - угрожать;
CHALLENGE - бросать вызов;
PERSUADE - убеждать;

BRIBE - подкупать;
JOIN - присоединяться;
GIVE TO - отдать кому-либо;
BEFRIEND - помогать;
INSULT - оскорблять;
WAIT FOR - ждать кого-либо;
TRACK - следить;

После выбора одного из приказов, Вам предложат выбрать, на кого этот приказ будет направлен. Это может быть определенный воин (THRALL, REAPER и т.д.) или какая-то обобщенная группа (FRIEND, ENEMY, ANYONE). Выбрав определенного воина, Вы должны еще и указать, какому лорду этот воин принадлежит. Таким образом, в конечном счете, приказ будет выглядеть примерно так: KILL UMBARG'S MYS- TIC или JOIN FRIEND.

Из этой группы можно выделить приказы BEWITCH, PETRIFY, CHARM и RELEASE. Ими может пользоваться только мистик. Эти приказы очень важны, поэтому о них мы расскажем подробнее:

BEWITCH - Воин, попавший под действие этого приказа, выходит из подчинения (Вы не можете им управлять). При этом он меняет свой цвет и получает такие приказы, как KILL FRIEND, KILL YOUR OWN REAPER и т.п.

PETRIFY - От этого заклятья воин окаменеваает. Он становится неспособным двигаться, но все еще сражается с встречными врагами.

CURSE - Попав под действие проклятья, воин становится слабым и трусливым, и его с легкостью убивает первый встреченный противник.

CHARM - действие этого заклятья мы не видели ни разу.

RELEASE - Мистик, имеющий этот приказ, может освободить воина от наложенных на него заклятий.

К сожалению, надо отметить, что вражеский мистик действует гораздо эффективнее Вашего, и к концу игры половина Вашей "армии" уже стоит окаменевшая или нападает на своих.

Замечателен также приказ PERSUADE. Он позволяет переманивать вражеских солдат на свою сторону (Естественно, это удается довольно редко). Лучше всего с данной задачей справляются персонажи с характеристиками MASTERFUL и PER- SUASIVE. О том, что воину удалось переманить врага, Вы узнаете по характерному звуковому сигналу.

Вторая группа. Приказы, направленные на предметы. Это:

GUARD - охранять;
SEEK - искать;
TAKE - взять;
GRAB - захватить, отнять;
USE - использовать;

После выбора одного из этих приказов Вам предложат выбрать предмет:

SWORD
AMULET
RING
SCEPTRE
BOOK
CUP
CRYSTAL
CROWN
ANYTHING

А затем еще и разновидности предмета (кроме того случая, когда вы выбираете ANYTHING):

GOLDEN - золотой
SILVER - серебряный
SHINING - блестящий
DARK - темный

Хотя предметов очень много, остается один глобальный вопрос: зачем они нужны??! Непонятно, как использовать команду USE. После выбора этого приказа, воин делает несколько шагов, и приказ исчезает. Возникает ощущение, что команда исполнена, но ничего не изменяется!

Третья группа. Приказы, оперирующие с картой.

BLOCK - блокировать проход
WAIT AT - ждать в определенном месте.

GO TO - идти куда-либо.

При выборе одного из этих приказов, Вам будет выдана карта острова, где будут показаны все Ваши воины, а выбранный воин будет еще и выделен яркостью. С помощью курсора Вы должны будете выбрать место назначения для воина.

В этой группе существуют еще два приказа, которые не требуют больше ничего выбирать. Это:
AMBUSH - устроить засаду;
ROAM - бродить.

Ну, и напоследок, осталось только описать управление. Клавишами "влево"- "вправо" Вы можете переключаться с одного воина на другого, а клавишами "вверх"- "вниз" переключать опции основного меню. Этим опций всего пять:

WATCH - позволяет незамедлительно переключиться на воина, участвующего в схватке. О том, что идет битва, Вы узнаете по звону мечей.

SCAN - показывается карта острова, с указанными на ней Вашими воинами. Выбранный воин выделен яркостью.

CHECK - показываются приказы, отданные воину, его качества и предмет, который он несет.

PLAN - вход в меню отдачи приказов.

QUIT - вход в меню, позволяющее загружать (отгружать) состояние игры, выйти в BASIC или продолжить игру.

В принципе, владея вышеизложенной информацией, в игру уже можно играть. Но вот выиграть у нас не получается. Поэтому мы обращаемся за помощью к читателям "РЕВИЮ", которым удалось разобраться больше - что же нужно сделать, чтобы победить?

ИФК: Надеемся, этот крик о помощи не останется без внимания, а мы пока переходим к следующему письму. Его прислал нам Андрей Назаренко из Рязани. Он уточняет, как заставить в ELITE пиратские корабли, которых Вы принудили к сотрудничеству, стрелять по противнику (см. ZX-РЕВИЮ 95/6, стр.55).

Корр: Я делаю так: Моих "помощников" я пропускаю немного вперед перед собой, или лечу с ними вместе. Как только на экране появляется объект, я торможу, а они летят прямо на корабль, развернувшись в его сторону. Мне остается только наблюдать, как они потрошат этого беднягу. После этого я лечу дальше, а они пристраиваются позади, а затем нужно вновь перестроится в боевой порядок.

Кстати, если у Вас есть Кобра-МК4, то Вы можете спокойно преодолеть любые пиратские или таргонские заслоны. Как только кордон появится перед Вами, поставьте прицел в центр планеты, и жмите J-двигатель вместе с одной из клавиш "влево"- "вправо". Ваш корабль будет так крутиться, что по нему никто не сможет попасть. Постепенно один кордон будет сменяться другим, пока Вы не долетите до станции. Таким образом можно легко долететь до двойных станций, выполнить миссию по взрыву станции, захваченной таргонами, и улететь от любого врага.

ИФК: теме ELITE посвящено и следующее письмо, которое нам прислал Сергей (фамилию свою он не указал) из Москвы.

Корр: Мне кажется, что бесполезно искать RAXXLA в нестандартных галактиках. Ведь авторы игры писали ее для западных пользователей, и наверняка не предполагали, что наши пираты начнут копать в отгрузочном блоке. Либо на RAXXLA можно попасть из стандартных галактик, либо ее нет совсем. Хотя, изучение нестандартных галактик тоже очень интересно.

В поисках Ракслы я проводил на ELITE M128 эксперимент по полету от звезды в пространство. Кораблей становилось все меньше и меньше, но они не пропадали совсем. После 1 часа 20 минут полета на J-двигателе ничего не изменилось. Причем, улететь от звезды оказалось гораздо труднее, чем приблизиться к ней. Стоило ненадолго развернуться в ее сторону, как она быстро приближалась, а удалялась медленно и мучительно. Беженцев мне впервые дали во второй галактике на планете ANAMA. В связи с этим возникают вопросы: А сохранилась ли эта планета? Если сохранилась, то здравствует, или до сих пор взрывается? А если не сохранилась, то где это зафиксировано?

Невидимые, да именно невидимые, а не двойные планеты нередко встречаются в нестандартных галактиках. Такие планеты не отображаются на карте и не обнаруживаются по "D" или "P". Возможно, это взорвавшиеся сверхновые или станции планет, не вступивших в Федерацию (вспомните Сираг в "The Dark Wheel").

Захватываемых таргонами станций (третья миссия) немного по одной в каждой галактике. В седьмой это DIAZEN, в восьмой ANESBI. В других галактиках мне миссию не давали. Вернувшись к прежним отгрузкам, я подгадал момент, и прилетел на DIAZEN как раз в тот момент, когда объявили "MAYDAY!"

Diazen Invaded!". Система планеты жила обычной жизнью, и никакими таргонами не пахло. Добавлю, что все это я проделал в Новосибирской версии.

Немного расскажу о поясе астероидов. За один раз у меня появляются до шести штук. Некоторые из них сопровождаются полицией, причем от остальных они не отличаются (ни внутри, ни снаружи). Некоторые астероиды начали меня сопровождать. Они подбирают (!) контейнеры и минералы. Если астероиды расстрелять, то тут же появится новая стайка. А если подождать, пока они отстанут, то можно немного полетать в одиночестве.

Совершенно случайно, в седьмой галактике, я обнаружил нелинейность карты. Из-за нее без дозаправки можно пролететь больше 7 световых лет. А именно:

ONERZARA-ANDIBIA : 6.0 L.Y.

ONERZARA-ATREXEXE : 8.0 L.Y.

ATREXEXE-ONERZARA : 7.6 L.Y.

ATREXEXE-ANDIBIA : 1.2 L.Y.

ANDIBIA-ATREXEXE : 2.0 L.Y.

ANDIBIA-ONERZARA : 5.6 L.Y.

Для полета от ATREXEXE до ONERZARA напрямик нужно 7.6 световых лет, а через ANDIBIA: 1.2 + 5.6 = 6.8 L.Y.! Возможно, это результат округления, но уж слишком грубого.

Любопытное наблюдение: торговцы, летящие навстречу "советуют", что везти с планеты, а пираты что везти на планету. (Я имею в виду содержимое их трюмов). Это не абсолютная истина, но иногда может помочь.

Если кто-то пытался высадить десант на своей старой Кобре-МК3, он наверное понял, что это нелегко. Прорваться сквозь заслон таргонов почти невозможно, стыковочный компьютер и спасательная капсула не помогают Вас захватят! Но если не удастся силой, надо брать хитростью. По моим наблюдениям через 2-3 сообщения "MAYDAY! Galactic 1 Invaded!" станцию планеты ESBEUS должны были захватить. Я лечу туда и жду. Причем, вблизи станции можно ждать сколько угодно похоже, время там просто не идет. Для того, чтобы оно шло, нужно отлететь со станции. Ее не должно быть видно, хотя сигнал "S" может присутствовать". Повисев на минимально возможном расстоянии, я дождался толпы таргонов. Лечу к станции таргоны отстали, и новые не появляются! Из станции вылетают, но очень вяло. Сразу после высадки десанта "RIGHT ON, COMMANDER!" и... целая стая: MAMBA, PYTHON и еще кто-то. Таргон их расстреливает (вот нахал!). Но мне пора в станцию...

ИФК: на этом Сергей заканчивает свой рассказ об ELITE, но его тему продолжает Алексей Горячин из г. Миллерово Ростовской обл.

Корр: Хочу поделиться опытом, и предостеречь остальных командиров от моей ошибки. Если Вы еще не приобрели "четверку" Кобру МК4, тоне выполняйте миссию MINERAL LIFE. Если же Вам натерпится (как натерпелось и мне в свое время), и Вы получите 35 тысяч кредов за минералы до покупки нового кораблика, то рискуете драться с таргонами на старом. Дело в том, что если отгружаться на каждой станции (а то вдруг убьют!), то примерно через 300 перелетов вам "подсунут" миссию ALLIEN COMPUTER. Хорошо, если удастся добраться до "правильной" планеты и скрыться, купив GALACTIC HYPERSPACE. Мне удалось купить "четверку" только в четвертой галактике.

Еще могу предложить способ для отыскания планет на главной диагонали. Находясь в первой галактике, вызовите ее карту (глобальную). Найдите (по "R") 2 планеты LERELACE на северо-западе и TYRAOR (или может TIRAOR, точно не помню) на юго-востоке, в центре. Приложите лист бумаги так, чтобы он делил экран пополам, а планеты будут служить Вам ориентиром. Не отрывая листа, перелетите в другую галактику. Вновь вызовите карту, и перепишите названия "подозрительных" планет, а затем посетите их лично.

А еще: известно ли Вам, куда летят "Вайперы", которые по каким-либо причинам не сопровождают Вас? Они летят вокруг планеты несут патрульную службу.

В поисках заветной Ракслы улетел в другой угол галактики. Лететь, раскачивая носом вверх-вниз, и удерживая включенным джамп, удастся лишь до тех пор, пока видна планета, а затем Вы вновь оказываетесь рядом с ней. Кстати, так можно "отбуксировать" Кориолис аж до астероидного пояса, сигнал станции уже давно пропал, а сама она вот, рядом крутится.

Напоследок, небольшая подсказка для тех, кто только-только начал свою карьеру. Когда Вы впервые (!) загрузили игру, то на LAVE нужно купить рабов и везти их на LEESTI. В результате Вы утроите свое состояние, правда, FUGITIVE получите тоже. К сожалению, такое возможно только единожды.

ИФК: Сразу несколько читателей откликнулись на просьбу о помощи Олега Смолянкина из г. Киселевск (см. ZX-РЕВИЮ N1-2'96). Вот что пишет по этому поводу Юрий Брагин из села Прелестное Белгородской обл.

Корр: Ответ можно найти в книге "Лучшие игры 1". Цитирую:

"... После того, как голова разлетится, на ее месте останется последний, пятый шар. Дорога на месте пропасти приведет Вас в маленькую комнату, в которой Вы найдете последний в этой игре ключ во времени. Забирайте его и идите к мертвому фараону. Рядом с фараоном есть камень, в который можно вставить Ваш ключ." Мне остается только добавить, что Вы, Олег, к сожалению, не на пятом уровне игры, а на четвертом. Вот названия всех уровней игры:

1. ROAD TO HELL - дорога в ад.
2. GREECE FOUR HUNDRED BC Греция, 400 лет до н.э.
3. SCANDINAVIA FIVE HUNDRED AD - Скандинавия, 500 год н.э.
4. EGYPT THREE SHOUSAND BC Египет, 3000 лет до н.э.
5. DUEL WITH DAMERON дуэль с Демероном.

ИФК: Николай Губин из Новгорода добавляет к этому, что из комнаты с ключом времени можно выйти, только присев (используйте первую опцию в меню).

ИФК: Андрей Сисюкин из г. Кисловодск уточняет информацию об игре ROBIN OF THE WOOD.

Корр: Во всех описания, которые я видел в этой игре есть неточность, вызванная, очевидно, желанием авторов совместить сюжет игры и одноименного фильма. Все пишут, что надо добыть три серебряные стрелы, а потом выиграть какие-то соревнования. Это не совсем верно. Действительно, три серебряных стрелы добыть надо, но никаких состязаний нет. Просто после того, как Вы меняете у дуба золото на меч, лук и стрелы, надо собрать три цветка и идти к фее. Она перенесет Вас в свой лес-замок. Там Вы найдете ключ от ворот и войдете внутрь. Буквально через пару комнат Вы увидите ворота, отличающиеся от всех остальных. Если Вы войдете в них, то получите благодарность, и сообщение, что Вы прошли всю игру. Мимо этих ворот пробежать невозможно, так как там всего один выход.

По ходу игры лучше начертить карту. Вся игра состоит из 5 (по моему) лесов, а каждый лес имеет 16 локаций на каждом ярусе, а всего ярусов в лесу около 20. Епископа можно встретить буквально в нескольких местах там, где на одном ярусе 3-5 локаций соединены в прямой "коридор".

ИФК: уточнение по игре HACKER 2 прислал нам Сергей Скиба (группа BEST HACKERS) из г. Угледар, Донецкая обл., Украина.

Корр: в ZX-РЕВИЮ 96/1-2 в разделе "Форум" на стр.103 была кратко описана игра HACKER 2. К сожалению, в описании есть неточность, а именно:

На Ваших роботах вообще нет видеокамер. Они установлены в коридорах базы. А Вы получаете с них изображение через спутник. К сожалению, то ли спутник устаревший, то ли помех много, но изображение все время мигает. Камеры разбросаны по всей базе и пронумерованы от 1 до 36. На один из мониторов в Вашей рубке Вы можете вывести изображение локатора (то есть небольшую карту) с помощью пиктограммы TGS. Другие пиктограммы:

SLCT выбор монитора.

MRU управление роботом. Поставьте стрелку на эту пиктограмму, затем нажмите "огонь". Пиктограмма меняет цвет - можете управлять роботом. За его перемещением следите по карте.

Да, кстати, чтобы не просматривать каждый раз при старте кучу малополезной информации, введите на запрос о пароле следующий код: 00987.

© А. Диденко, 1996 г.
"ЗВЕЗДНОЕ НАСЛЕДИЕ"
STEP 1995 г.

Сразу после приземления осмотрите спасательную шлюпку. Включите бортовой компьютер, и попросите его оказать первую помощь. Используйте ампулу и идите на север. Помните, что в некоторых местах отдыхать смертельно опасно!

После того, как убьете пантеру, идите на восток и убейте артанга. Обыскав труп и амортизационный кокон, Вы найдете осколок стекла, таблетки и нейрохлыст. После этого возвращайтесь к месту приземления. С помощью осколка отсоедините аккумулятор от приборной панели.

Зарядите нейрохлыст, после чего отправляйтесь на восток. Не ходите ночью по болоту! Убейте "водяного", сделайте еще один ход на восток, после этого поверните на север и убейте незнакомца. Заберите у него бластер и кредитку и возвращайтесь на юг. Идите на восток до тех пор, пока не придете на заводской

двор. Выложите бластер и нейрохлыст и приподнимите лист железа. Вы увидите подземный ход. Пройдя по нему, Вы окажетесь на берегу моря. Идите на запад, пока не увидите винтолет. Прodelайте с помощью бластера дыру в люке и снимите со скелета браслет. Пройдя еще на запад подкрепитесь ягодами.

Не ходите дальше! Вернувшись к месту, где вы убили человека, идите без отдыха на восток, пока не придете к дому старухи. Отдайте ей браслет и отдохните в ее доме. Старуха даст Вам пропуск на глайдер и записку для Джеймсона. Сделав один ход на запад и два на север, Вы окажетесь у лавки Джеймсона. Зайдите туда и отдайте хозяину записку от старухи и медальон. Джеймсон скажет, что он хорошо знал Вашего отца и отдаст Вам клочок бумаги с цифрами и бутылку ликера. Выпив ликер и разбив бутылку, Вы обнаружите кусочек пластика с цифрами. Попытайтесь снять картину, висящую в лавке. На ее обратной стороне Вы прочитаете стихотворение. С помощью цифр на куске бумаги и пластика составьте две фразы - "ГОРА ПОЛДЕНЬ" и "СЛОВО САМ". Теперь идите к месту Вашей схватки с незнакомцем и сделайте оттуда два хода на запад. Вы окажетесь у скалы. В полдень откроется вход в пещеру. Войдите туда, вставьте в прорезь под терминалом кусок пластика и введите слово "САМ". После этого компьютер сообщит Вам адрес, по которому Вы должны доставить футляр. В нише Вы обнаружите футляр и биомаску. После этого отправляйтесь в порт, который находится рядом с домом старухи. Не забудьте надеть биомаску! Предъявите капитану глайдера пропуск и отправляйтесь на материк.

На этом первая часть ваших приключений закончена. В дальнейшем я буду менее подробно описывать свои перемещения, чтобы не лишать игроков удовольствия.

После прибытия на материк Вы обнаружите, что у Вас пропал футляр, поэтому первым делом, узнайте, куда направляется капитан, и идите вслед за ним в бар. При разговоре с капитаном не следует снимать биомаску! В ходе разговора выяснится, что капитан видел, как какой-то человек вытащил у Вас футляр. Капитан отдаст Вам электронный ключ, потерянный вором. После разговора некоторое время походите в окрестностях бара. Скоро Вы наткнетесь на двух инопланетян. Убейте их и заберите монету. При этом Вам придется поманипулировать с биомаской. Затем идите на набережную. По пути можете зайти в здание порта и посмотреть интересный сон. На набережной в беседке Вы увидите мусорную кучу. Основательно покопавшись в ней, Вы найдете бумажку и кусок проволоки. Теперь отправляйтесь в аэропорт. Там, послушав новости, можно узнать немало интересного и полезного, а после этого покупайте билет до Тарана и в путь!

В Таране, неподалеку от дома, в который Вы должны отнести футляр, можно встретить старика, который расскажет Вам про Темпор. После этого он умирает, а Вы сможете найти у него пакетик препарата "Брейк", который, как я понял, используется для повышения Вашей энергии. На одной из окраин Вы увидите автокар. Зайдите в него и откройте дверь в кабину с помощью проволоки. В кабине Вы найдете пропуск. Возьмите его и идите к КПП.

Пройдя КПП и подкрепившись по дороге ягодами, пройдите к даче. Спрячьтесь за кустами, которые растут около входа и подождите. Скоро появится и сам хозяин дачи, который, как мы знаем, потерял электронный ключ. Убейте его и обыщите. В результате, Вы найдете похищенный футляр и документы на имя Арбеса Вереса. Теперь открываем дверь, предварительно вытащив из замка записку, и заходим в дом. Не забудьте запереть входную дверь! Берем часы и запоминаем картину "Врата Темпора". Эта картина очень поможет нам в дальнейшем. Затем идем в кабинет, где стоит сейф. Читаем надпись на электронном ключе: "КОГДА ОСТАНЕМСЯ НАЕДИНЕ: ПЕРВЫЕ И ВТОРЫЕ". Сложив первые и вторые буквы получаем два слова: "КОН" и "ОСА". Поэкспериментировав с электронным ключом и часами, обнаруживаем, что если приложить ключ к часам, то на них меняется изображение. Установив на часах изображение пчелы, вставляем их в прорезь на сейфе. Набираем на шифронаборнике слово "КОН", и...сейф открыт! Заберите из него пропуск до Тарана, вытащите из прорези часы и выходите из дома, еще раз посмотрев на картину. Теперь обратно в Таран через КПП.

Как только Вы окажетесь в Таране, сразу идите по адресу, который Вам сообщил компьютер на острове О. Осмотрев квартал и подойдя к боксу 12, позвоните и, вслед за этим, приложите к глазку сначала футляр, а затем медальон. Дверь откроется, и Вас пустят в квартиру. Перед тем, как позвонить, необходимо снять биомаску! Вы отдадите футляр, а взамен получите электронный навигатор и специальный бинокль. После этого идите к супермаркету и обменяйте у нищего часы на пакетик препарата "Брейк". Теперь отправляйтесь на аэродром, дождитесь, когда там приземлится челнок "Плот", залезайте в кабину, вставляйте электронный навигатор и летите... в Аркан. По пути Вас собьют, и Вы снова окажетесь на острове О. Пройдя по берегу моря на запад, Вы увидите винтолет и двух карликов, которые сразу на Вас набросятся. Убив их и забрав копьё, бульжник и коробку, идите на восток. Покопавшись копьём в обломках челнока, Вы найдете электронный навигатор. Теперь идите на запад до того места, где растут ягоды. Поешьте и разбейте коробку бульжником. Вы станете обладателем прибора "Оборотень-7". Включите его и смело идите ми-

мо толпы карликов к "Кобре МК-3". Входите в корабль и, убив паука, летите на шлюпке "Муха" в Темпоральную Зону.

Приземлившись, вспомните, что было изображено на картине "Врата Темпора". Принимаем цифры за стороны света: 1-север, 2-восток, 3-юг, 4-запад. Число 'пи' равняется 3,1415... Идем на юг, на север, на запад, на север. Тут возникает проблема: цифра '5' явно не соответствует ни одной из сторон света. Методом "научного тыка" удалось выяснить, что нужно сделать еще один ход на восток. Посмотрев в бинокль, Вы увидите огромный сгусток. С его помощью Вы пройдете игру "ЗВЕЗДНОЕ НАСЛЕДИЕ".

Ну, вот и все. Прошу извинить меня за возможные неточности, так как писал по памяти.

ИФК: Об игре COLONY рассказывает Павел Козлов из г. Асбест Свердловской обл.

Корр: На небольшой планете обнаружили грибы, обладающие необычайными свойствами. Из экстракта этих грибов можно сделать эликсир, продлевающий людям жизнь. Так как ни на какой другой планете эти грибы не росли, то землянам пришлось основать здесь свою колонию. Из-за тяжелых природных условий люди на планете не жили, а вся колония управлялась дроидом, который получал команды с Земли.

Дроид был предназначен не только для выращивания и отправки грибов на землю, но и для охраны их от местных вредителей - термитов и жуков.

В ведении дрoида находился комплекс, который включал в себя три склада с оградами, один склад с органическими запасами, генератор, места для грибов, панелей и общие запасы.

Теперь перехожу к самой игре:

Вначале Вам предстоит выбрать управление, затем нажимайте "огонь" - и вперед! Когда игра начнется, Вы увидите комплекс. Он огражден забором, но после атак всякой живности, от него ничего не останется. У Вас есть возможность пополнять запасы в следующих складах:

STEEL FENCE STORE - склад стальных секций к ограде.

BARBED WIRE STORE - склад колючей проволоки.

FENCING STORE - склад оград из дерева.

Кроме того, у Вас есть:

ORGANIC SUPPLIES - органические запасы, где Вы можете:

DEPOSIT MUSHROOM - сдать грибы;

COLLECT SEEDS - набрать семена;

DEPOSIT SEEDS - сдать семена.

Также, здесь показывается:

MUSHROOM PACKED - общее количество упакованных грибов.

SEED IN STOCK - количество семян на складе.

Помимо этого, в Вашем распоряжении GENERAL SUPPLIES (основные запасы):

DEPOSIT PANELS - сдать солнечные батареи;

COLLECT PANELS - взять солнечные батареи;

COLLECT TRAPS - взять ловушки (для жуков);

DEPOSIT TRAPS - сдать ловушки.

EXIT - выход из GENERAL SUPPLIES. Здесь Вы, также, можете узнать, сколько батарей и ловушек осталось на складе.

Последнее здание - центр управления (CONTROL CENTRE). Здесь Вы можете включить маяк (ACTIVATE BEACON) и отключить его (DEACTIVATE BEACON), а также заказать себе товары с Земли:

название	количество в коробке	цена в CR	для чего предназначены
DROID BATTERY (батарея дрoида)	1	200	для заправки второго дрoида (красного)
REMOTE DROID	1	1000	?
SHARE DROID BODY	1	1000	?
HEAVY DUTY LASER	1	500	для уменьшения затраты энергии лазера
MEGA FORCE FIELD	1	500	для уменьшения затрат энергии дрoида
DEATH TRAP (ловушки)	10	30	ловушки для жуков
STEEL FENCE (стальная ограда)	10	100	-
BARBED WIRE (колючая проволока)	10	60	-
WOODEN FENCE (деревянная)	10	80	-

ограда)			
EXTRA SEEDS (семена грибов)	10	50	для выращивания грибов
SOLAR PANELS (солнечные батареи)	10	100	-

ИФК: Андрей Базникин из п.Орджоникидзе, Красноярского края задает вопрос об игре TOTAL ECLIPSE 2.

Корр: В этой игре я собрал 10 из 14 частей сфинкса и застрял в комнате с невидимыми стенами. Что мне делать дальше?!

ИФК: Мысль об усовершенствовании знаменитой ELITE не дает спокойно спать космическим асам с разных концов галактики. Неудивительно, что наша идея собрать определенную "базу данных" в помощь программисту-новатору вызвала такой большой читательский резонанс. На сегодняшний день "база данных" практически полностью собрана, а завершает ее серьезный труд синклеристов с Украины, переданный нам Александром Баженовым из г. Кемерово.

Корр: Напомню, о чем, собственно была эта статья, там предлагалось расширить параметры игры до захвата кораблей (абордаж), с которых уже катапультировался экипаж, а также о покупке частных станций, или, скажем, устаревших космических платформ и т.п.

Кроме боевого захвата можно предложить и мирный способ, например, денежный займ правительству планеты, спонсирование госпредприятий, вкладывание инвестиций в строительство и экономику лишь бы втянуть руководство этой планеты в "долговую яму". А когда сумма долга превысит определенную отметку (например, годовой денежный оборот этой планеты), то правительство в Ваших руках, эта планетка Ваша!!! Заманчиво ведь?

Пойдем дальше... "Планета Ваша" это, конечно, громко сказано, никто и не собирался устраивать там государственный переворот. Но отказать Вам в предоставлении своего военного или грузового флота (если у них там таковой имеется) они уже не смогут. Для них это будет чем-то вроде погашения процентов или отсрочки сроков выплаты. А уж как Вы будете использовать взятый "напрокат" флот это сугубо личное дело каждого пилота.

Разбить галактики на Звездные королевства, Союзы, Советы и обязательно ввести между ними "напряженку" в дипломатических отношениях, а можно и локальные войны. Такая игровая обстановка поможет играющему развить навыки дипломата. Можно разделить программу, отделив летную часть от торговой и стратегической пусть она будет подгружаемой, как "Звездное наследие" или "НЛО". Перелетел в другую галактику заработал дисковод, и нужная информация на экране. На станцию садишься снова дозагрузка. Такая система позволит освободить в ОЗУ немало пространства для творчества. Можно улучшить звуковое оформление, по желанию включать и выключать музыку, и, если надо, довести количество галактик до 3-4.

Есть очень нужная доработка, которую нужно бы ввести в первую очередь это счетчик игрового времени. Ведь почти любой на вопрос: "Сколько ты летаешь?" сможет ответить лишь приблизительно. Еще можно увеличить число членов экипажа хотя бы до двух человек. Давайте-ка вспомним фантастическую повесть... Там присутствует момент, когда герои ведут бой вдвоем. И еще, уважаемые пилоты, вы не могли не заметить, что противник умудряется вести огонь, даже не находясь с Вами в одной плоскости, причем стреляет он довольно точно! Мы полагаем, что здесь можно найти компромисс, роль стрелка может выполнять второй игрок или бортовой компьютер, который, в свою очередь, можно усовершенствовать, купив за кругленькую сумму "прицельно-логический" блок. Прицел должен быть подвижным, иметь возможность закрепляться в любом месте экрана, а по нажатию определенной клавиши (комбинации клавиш) выставляться точно по курсу корабля.

Можно было бы ввести новые корабли, например, "Летучий Голландец" с какой-нибудь заразой на борту, корабли ловушки, корабли давно исчезнувших цивилизаций с автоматической системой охраны. Такой корабль не так-то просто присвоить, атаковать его равносильно самоубийству. Зато, прицепив его к себе, можно получить невероятное оружие, компьютеры или... смертельную дозу неизвестного излучения. Тут уже появляется примесь адвентюры: хочешь, лезь внутрь, а хочешь, взорви и лети дальше.

Модифицировать старенькие корабли тоже можно. Оборудование это вообще целое море возможностей. Начиная с запрещенного полицией определителя "свой-чужой", который есть на VIPER'e и FER-DE-LENCE до фальшивых электронных паспортов для пиратствующих кораблей. Это все можно приобрести на "черном рынке", речь о котором пойдет дальше, или брать с захваченных кораблей.

Также, большой темой является вопрос: откуда берутся многочисленные полчища Таргонов. Нельзя ли, наконец, рассекретить эту информацию, так упорно скрываемую программой. Может быть, пора про-

демонстрировать нашу силу и заставить их пойти на переговоры или разбить их в пух и прах! Может быть, только так мы найдем планету-легенду. Там, где ее никто не додумался бы искать...

Теперь перейдем к станциям и торговле. Торговать, конечно, можно и, не выходя из корабля, но непонятно, как можно побывать на станции, и не посетить какой-нибудь бар или закусочную. Там можно отдохнуть от трудов праведных (и неправедных), расслабиться, встретить знакомых, да просто поболтать и поделиться новостями. Если позволит правовой статус прогуляться по "Черному рынку". При чем здесь статус? Да при том, что Вам никто не скажет про черный рынок, если Вы "Чист" (значит, точно "стукач"). Да где это видано, чтобы запрещенными товарами торговали через Главный Коммерческий Компьютер!? Вас после такой операции арестуют еще до выхода со станции. Другое дело инкогнито загрузить контейнеры с оружием или наркотиками, купленными на "черном рынке" по липовым документам. Кстати, полицейские досмотры, если Вы встретите VIPER'ы в открытом космосе, должны стать обычным делом. Очень прибыльно было бы запрещенные товары доставлять прямо на планету, минуя полицейский контроль и таможенную. Если COBRA сама сесть на поверхность не может, можно специально для этих целей таскать с собой 1-2 SIDEWINDER'a или ADDER'a. В фирменном описании значится, что эти корабли могут летать над поверхностью и совершать посадки. Допустим, "Кобра" в автоматическом режиме висит где-нибудь вне зоны досягаемости средств ПВО, а Вы, на небольшом катере, пытаетесь незамеченным прорваться к северному полюсу какой-нибудь планетки, где среди вечной мерзлоты спрятана база старого космического волка. Координаты ее ни в каком справочнике не записаны, их можно узнать только на самом "черном рынке".

Вот, вкратце, и все основные идеи. Будем уповать на наших замечательных программистов, и ждать "SUPER ELITE".

ИФК: Женя Сивоконь (возможно, ошибка) из г. Горячий Ключ Краснодарского края задает вопрос по игре VENTURAMA.

КОРР: После того, как лифт вывез меня из пещеры на твердую поверхность, я ничего не смог сделать. Я нашел RUSTY OLD KEY и подошел к сломанному мосту. Дальше мне пройти не удалось. HELP.

ИФК: Вообще, все виденные нами версии этой игры были с ошибкой именно в этом месте, по экрану ползла всякая гадость, затрудняющая игру. Так что если у Вас такая же ситуация, то пройти дальше не удастся... Или, может быть, кому-то это удалось?

ИФК: Наш корреспондент из г. Асбест Свердловской обл., Денис Ефремов (DIN SOFT) отвечает на вопросы по игре CARRIER COMMAND.

КОРР: В "ZX-РЕБИО N1-2 1996" Илья Командин из г. Калуга спрашивает: "Почему после установки фабрики или защитного комплекса не всегда отстраиваются необходимые сооружения (хотя на карте в режиме RESOURCE они видны)? Нельзя ли на таком острове установить одновременно фабрику и ресурсы?"

Отвечаю: Для того чтобы наблюдать за строительством объекта, который Вы благополучно выкинули на остров, необходимо отменить все существующие заказы в режиме планирования количества оборудования. Надо оставить одни нули.

Если этого не сделать, то компьютер сначала выполнит Ваш заказ на производство нужного Вам оборудования, а потом приступит к строительству на захваченном Вами острове фабрики, вышки или аэродрома.

Ответ на второй вопрос: На один остров нельзя десантировать два объекта. Так, например, ни фабрика, ни буровая вышка, ни аэродром на одном острове существовать не могут. Компьютер пишет что-то вроде "Здесь уже построен другой объект".

Еще совет: Когда Вы летите на MANTA бомбить вражеский ангар на острове, добирайтесь до него на максимальных скорости и высоте этим Вы избежите, поражения ракетой, а если Вы поставите авианосец достаточно далеко от острова (седьмая риска на эхолоте), то вражеские ракеты не смогут его поразить.

Еще совет: самый ответственный и переломный момент в игре это встреча с авианосцем STANZA. Рекомендую эту встречу производить рядом с Вашим островом, на котором расположен аэродром и вулкан. От Вас потребуется выпустить защитных дроидов (чтобы сбить вражеские ракеты и помешать противнику приблизиться). Свой авианосец нужно спрятать за вулкан. Аэродром, располагающийся на острове, сам разберется с врагами.

ИФК: Следующее письмо также пришло к нам из г. Асбест. Павел Козлов сообщает читателям новую, информацию по SIM SITY и дает довольно нетрадиционные советы:

Корр: Полицейские участки лучше никуда не ставить, они отнимают деньги из бюджета и затрудняют (!!!) развитие города. Кстати, пожарные депо тоже лучше не ставить огня они не тушат, зато горят horribly.

Аэропорты лучше ставить в районы с большим количеством жилых построек это способствует быстрому заселению последних. То же самое можно сказать и про индустрию и коммерцию.

Если Вас сильно беспокоят пробки (TRAFFIC) на дорогах, и Вас совершенно не волнует внешний вид города, то дорогу лучше строить "обрывками", то есть через клетку. Это никак не затрудняет развитие построек и экономит деньги на оплату графы TRANS в городском бюджете. А вообще, лучше не мучаться, и строить железные дороги.

ИФК: Виктор Комаров из г. Петропавловск-Камчатский сетует на то, что в нашем журнале нет ни слова об игре FIRELORD. Он просит рассказать о ней подробнее.

ИФК: Алексей Семенов из Рязани просит экспертов написать в журнал об игре THINK фирмы TIGERLESS SOFTWARE.

Корр: эта игра является оригинальной переработкой "крестиков-ноликов". На игровом поле размером 6х6 клеток надо выстроить 4 фишки в ряд. Ставить фишки нужно на правый или нижний ряд. При этом соответствующий ряд или колонка сдвигаются. Однако, есть некоторые ограничения по возможности хода. Может, кто-то знает больше?

Корр: Также, меня интересует, чьи потери показываются в стычках в стратегических играх (STALINGRAD, WATERLOO...). При боях между разными армиями возникают цифры. Что конкретно они обозначают потери компьютера, или меня?

ИФК: Небольшие трики в игре RAMPAGE описывает Александр Женинов из с.Троицкое, Алтайского края.

Корр: Самое простое, что можно сделать это подойти к правому краю экрана и бить кулаком вправо. Машины, выезжающие или подъезжающие к левому краю экрана будут взрываться. Точно так же можно взрывать и вертолеты для этого надо прыгать.

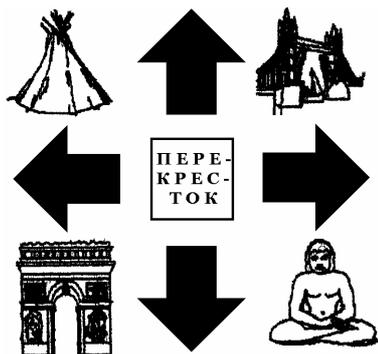
Если подвести одного героя к краю экрана (спиной к домам), а другим спрыгнуть с дома в сторону первого, а первым ударить кулаком вверх, то тот, который прыгнул, останется висеть в воздухе, и даже сможет уйти за экран.

© Д. Анисимов, г. Киров, 1996

Хочу сказать пару слов об "А.Т.Ф.". В описании этой игры (ZX-РЕВЮ за 1993 г., стр. 257) не указана клавиша "G". При ее нажатии выбирается ближайший, из текущего типа целей, объект. В ZX-РЕВЮ 94/3 на стр. 65 написано, что автопилот в некоторых случаях не работает, и самолет пролетает аэродром. Я с этим не согласен. Дело в том, что совершить посадку можно, когда азимут самолета и аэродрома равны 0, 90, 180 или 360 градусов. И поэтому автопилот производит серию маневров для достижения такого азимута. А если автопилот включить слишком близко к аэродрому, то он не успевает совершить маневр, и самолет пролетает аэродром.

Однако, пролетев некоторое расстояние, самолет разворачивается и удачно совершает посадку.

ПЕРЕКРЁСТОК



Олег Бобрышев из г. Георгиевск продолжает исследование игры HE-MAN. Он выяснил, что в зазеркалье есть камень седловидной формы (STIRRUP-SHAPED BOULDER). При его обследовании нам сообщают, что это сильная (хорошая) поддержка (IT'S A STRONG SUPPORT). Возможно, прежде чем спускаться, надо что-нибудь сделать с ним?

Также, насчет CUBOID'a. Если его исследовать, нам скажут, что у него "поверхность с малым трением, невозможно ухватиться". А о мхе говорят: "он прилипает к Вашим рукам". Стало быть, приходим к кубоиду со мхом и проблем не будет.

Также Олег рассказывает о своих приключениях в программе HAMPSTEAD, которую он прошел почти полностью.

Кстати, чтобы взять LATHE RETAINING BRACKET, надо в промышленном районе двигаться по определенному маршруту. Когда Вы попадете в этот район, идите так: E-E-NE-E-E-NE-N-SW

Далее, прибыв в Лондон, можно зайти в офис агента по недвижимости. Обследовав предложения (EXAMINE DESCRIPTION) мы узнаем, что здесь продается коттедж с видом на Хэмпстедскую Степь. Всего за 100000... Пока это не для нас.

Кроме того, в магазине, в мебельном отделе, есть маленький письменный стол. Открыв его, мы обнаружим весьма ценную вещь отвертку! (A STURDY SCREWDRIVER). Вот теперь, с этой отверткой, мы можем вскрыть шкаф в своем офисе (FORCE CABINET WITH SCREWDRIVER) и обнаружить там очень интересный конфиденциальный доклад по ценам в Сан-Пауло.

1. Повышение цен приведет к доходности меньшей, чем чистые расходы.
2. Статичные цены не помогут при повышении реальной стоимости.
3. Снижение цен улучшит скупку проекта.

Сопоставив эти данные с изученным ранее меморандумом, мы уже знаем, как проголосовать, и на совещании выбираем OPTION 3.

После совещания директор поздравит Вас с началом карьеры, вручит банковское обязательство на 100000 (где-то мы уже видели эту цифру). Надо заехать в офис по продаже недвижимости приобрести себе жилье. Кстати, в коридоре банка висит девиз компании "PER ARDUA AD NAUSEAM" (видимо, по аналогии с PER ASPERA AD ASTRA, "Через тернии к звездам"). Переведя эту надпись (TRANSLATE), мы узнаем, что она значит "Через работу к болезни" (аноним).

Купив коттедж (PURCHASE COTTAGE), отправляемся на вечеринку. Теперь фотомодель Пиппа более благосклонно к Вам относится, и ее можно взять с собой (TAKE PIPPA). Кстати, не стоит сразу же пытаться делать Пиппе неприличные предложения. Во-первых, кругом люди, а во-вторых, "это не тот язык, на котором говорят в Хэмпстеде", как Вам скажут вместе с GAME OVER.

Теперь в Лондоне делать уже нечего, так что с Пиппой на руках идем к автомобилю. (Вы, наверное, носите ее на руках, так как она появляется в INVENTORY, и учитывается, как и любой другой предмет). В этот момент лучше иметь с собой и меморандум и доклад. Это означает, что все, кроме прав на коттедж, меморандума и доклада, а также, естественно, Пиппы и машины, придется выкинуть.

На машине выезжаем на дорогу, и тут Пиппа сообщает, что мы должны посетить ее отца, которого зовут LORD FISH, но которого все знают, как Чабби (Крепыш).

Она добавляет, что Чабби ужасно важен в правительстве и является политическим противником сэра Лионеля Трампа.

Проехав дальше, попадаем на перекресток. Отсюда можно ехать в St. JOHN'S WOOD (Лес Святого Джона), HAMPSTEAD, CAMBRIDGE и ISLINGTON.

В Хэмпстед ехать пока рановато, при поездке в Кембридж у нас на полдороге кончается горючее, и, как следствие, GAME OVER. Отправившись в ISLINGTON, мы увидим, что дорога разветвляется на две все в те же Кембридж и Хэмпстед. Остается лишь один путь.

По дороге в лес Святого Джона у нас вновь появляется выбор можно ехать к внушительному помещью 19-го века, либо в бедный район, откуда мы начали игру по той самой односторонней дороге. (И, так как движение одностороннее, обратно по ней уже никуда не уедешь, да и, вообще, не выбирешься обратно. Поэтому сюда лучше не ходить.

Отправляемся в помещье. Здесь колокольчик, а над дверью геральдический щит. На щите изображена рыба, стоящая на задних плавниках. Очевидно, помещье принадлежит лорду FISH. Звоним в колокольчик (PULL BELL). Впрочем, по звукам можно скорее сказать, что мы имеем дело с парходным гуд-

ком. Дверь открывает слуга, интересующийся сущностью нашего дела. Очевидно, нам надо повидать Чабби, так и отвечаем (MEET CHUBBY). Слуга приглашает нас войти. Внимание! Здесь, главное, не забыть оставить машину. Машина не велосипед, так что при попытке въехать на ней в любое помещение, Вы получаете GAME OVER.

Войдя в поместье, Вы встретитесь с Чабби. Он сердечно Вас поприветствует и, после начальных любезностей, поинтересуется о Ваших намерениях по отношению к его дочери. Единственный правильный ответ здесь жениться (MARRY PIPPA). Кстати, любой другой ответ приведет к тому, что Вас вышвырнут, а назад Вы уже не вернетесь.

Удовлетворенный, Чабби щедро наливает Вам скотча. Пиппа рассказывает ему о Вашем недавнем успехе в банке. Интерес Чабби растет. Он хочет больше узнать о проекте Сан-Пауло.

Здесь ему нужно дать меморандум (GIVE MEMO TO CHUBBY). Лорд Фиш интересуется, нет ли у Вас еще чего-нибудь. Отдайте ему доклад (GIVE REPORT TO CHUBBY). (Кстати, документы можно давать в любой последовательности).

Прочтя новый документ, лорд Фиш оживает. Он сообщает, что дни сэра Лионеля сочтены. А так как действия в Сан-Пауло приведут его к потере поста в правительстве, то этот пост Ваш, вместе с обязательным рыцарством! Вот так просто свалить своего начальника. Тут входит слуга. Он напоминает лорду о заседании Кабинета, на котором требуется его присутствие. С наилучшими пожеланиями Чабби оставляет Вас.

На этом достижения Олега заканчиваются. Все вышеперечисленные действия "честно" дают 95%. Но проценты можно добыть и "нечестно" каждый перевод надписи дает по два процента. То же касается и Пиппы. Когда Вы ее берете с собой, получаете 5%. Если ее бросить и снова поднять (вот это зрелище, наверное), Вы получите еще 5% и так до бесконечности.

Поток писем по игре **ЗВЕЗДНОЕ НАСЛЕДИЕ** и не думает прекращаться, несмотря на то, что мы вроде бы уже осветили все основные моменты этой программы. Иногда нам попадаются и такие письма, в которых авторы предлагают вообще не отправляться в Таран, а купить в аэропорту билет до Аркана и закончить игру.

На самом же деле, это ответвление сюжета с Арканом введено авторами лишь для создания соответствующей сюжету атмосферы, и оно тупиковое. Поэтому, разумеется, говорить о том, что "я прошел игру не мучаясь взял билет и улетел в Аркан" несколько некорректно ведь вселенная остается под гнетом артангов, и главная цель игры переход через темпор не выполнена.

Продолжается мозговой штурм на программу REBEL PLANET. Свою лепту в общее дело вносит Леонид Хмелинин из Екатеринбурга. Он выяснил, что надо делать на второй планете.

Как выяснилось, в отеле надо попросить HOTELIER'a (ASK HOTELIER FOR ZODDI SPECIAL), а потом в комнате 110 можно взять одеяло. Имея одеяло, мы сможем избавиться от робота на станции HALLIS, в подвале музея. Для этого надо бросить одеяло на его антенну THROW BLANKET ON ANTENNA. После этого возьмите кристалл и проволоку. В DESCRIPIT EAST-WEST PASSAGE идите на восток, теперь надо забрать акустическую вилку (PHONIC FORK) там, где Вы ее оставили, и вместе с кристаллом и вилкой поезжайте на станцию YOTO. Около музея надо подождать (WAIT), а потом постучать вилкой по кристаллу (TAP CRYSTAL WITH FORK). В университете возьмите кусачки. Кофе пить не надо! С кусачками, проволокой, JET PACK'ом и PANTANIUM POWER PAK'ом надо ехать до станции TUNDRA. Подсоедините PAK к забору (CONNECT PAK TO FENCE) и разрежьте его (CUT FENCE WITH CUTTERS).

В тундре, изучив скалу в виде купола, возьмите TIN и подсоедините его к ранцу FIT TIN TO PACK. Оденьте его, и в локации с широкой рекой и двумя пиками, переправьтесь через реку (TUG JOYSTICK). CRAG SNAPPER'a надо отогнать командой SING. Не стоит изучать маленькую трещинку, лучше обратите внимание на большую. Там лежат следующие вещи HCAP, LUMP OF SHINY METAL, MIND PROBE и BROKEN SPY MISSILE. Взяв пробник, перепрыгиваем через реку, а потом прыгаем еще раз из той локации, где виден маленький домик. Оказавшись перед домиком, надеваем PROBE. Дальше, если пойти наверх, мы встречаем человека по имени DORADO, его еще нам советовали найти в университете. Если с этим DORADO поговорить, то он спросит у Вас пароль. А вот какой этот пароль, Леониду узнать не удалось.

Также, ему не удалось узнать как открыть OCTOPLAST DOOR на первой планете, как убить HAL-WOLF'a и что делать с призраком (SHADOW)? Есть предположение, что тут может пригодиться библия из комнаты 110 в отеле.

А.П. Бабкин и И.В. Зуенок из г. Магадан освещают две малоизвестные адвентюры:

Первая из игр, под названием **THE FAMOUS FIVE**, выпущена не очень известной фирмой ENIGMA VARIATOINS и основана на детских детективах, герои которых, четыре школьника и собака. После запуска игры появляется краткая предыстория, которую нет смысла здесь пересказывать.

В начале игры Вы, как JULIAN (между героями можно переключаться командой BECOME <имя героя>), едете в поезде вместе с братом DICK'ом и сестрой ANNE. У DICK'a есть с собой немного денег, а у ANNE сэндвичи, завернутые в бумагу.

Через некоторое время поезд останавливается, и будет логично выйти (LEAVE). На платформе Вас встречает тетя FANNY и предлагает следовать за ней. Таким образом, Вы приходите в дом, который не мешало бы обследовать.

1. На чердаке есть окно, которое можно открыть, но зачем неясно.

2. В комнате дяди и тети стоят часы, из которых можно вытащить батарейки (TAKE BATT).

3. В SPARE ROOM под кроватью лежат спички (EXAM BED).

4. В гостиной можно включить радио (TURN SET), и послушать прогноз погоды. При этом придет кухарка JOANN'a и выключит радио, мотивируя это тем, что дядя QUENTIN работает, и ему это мешает. У нее надо попросить прихватки (SAY TO JOANN GIVE ME GLOVES, причем все набирается без кавычек).

5. На кухню просто так не пройти выгоняет JOANN'a. Чтобы попасть в сад, расположенный за кухней, надо оставить семейку в столовой, переключиться на GEORG'a (это ребенок тети FANNY). Как GEORGE, пройдите в гостиную, и включите радио, а потом тут же переключитесь на кого-нибудь из троицы. Пока JOANN'a будет разбираться с GEORGE'м, Вы сможете спокойно пройти в сад. Также можно пройти в кладовку, расположенную на юго-восток от кухни.

6. Темную комнату можно осветить спичками (LIGHT MATCH). В ней лежит фонарик. Вот тут-то и пригодятся батарейки (PUT BATT IN TORCH).

7. В кабинете дяди много научных книг. В одной из них написано что-то вроде (READ BOOK): "Возрастающая скорость, вызванная уменьшением размеров, зачастую вызывает разрушение материи любой прочности". К чему бы это?

8. В сарае лежит веревка. В него можно попасть не только через улицу, но и через кабинет FRENCH WINDOWS это застекленная дверь.

9. Чтобы подружиться с GEORGE'м, надо купить в магазине COOL ICES и дать ему.

10. На берегу, сын рыбака ALF возвращает GEORG'у собаку.

11. В лодку можно сесть командой IN, потом надо взять весла и грести (ROW BOAT). Причем грести должен GEORGE.

12. На полпути к острову в воде виден затонувший корабль, к нему можно нырнуть (LEAVE BOAT, DOWN). Но войти в корабль не получилось по двум причинам: во-первых, не хватает воздуха, а, во-вторых, очень опасно.

13. В бухте вытащите лодку на берег (PULL BOAT), чтобы не унесло.

14. На восточном конце острова есть дыра. Чтобы в нее спуститься, надо привязать веревку к пню (EXAM NEATHER, TIE ROPE TO STUMP), а потом бросить ее в дыру (DROP ROPE). Теперь DOWN.

15. В пещере лежит деревянный ящик, заколоченный гвоздями (CRATE). Сломать его может только JULIAN.

16. Во внутреннем дворе замка собака бросается за кроликом, и проваливается под землю. Чтобы ее достать, надо надеть прихватки и выдернуть куст (PULL BUSH). При этом откроется ход под землю. Дальнейший ход вниз завален плитой, но все же остается очень узкое отверстие, в которое может пролезть только DICK. Здесь уже темно, и необходимо включить фонарик (TURN TORCH) Вы увидите, что лестница обрывается, а путь продолжается дальше. Привязывайте веревку к лестнице и можете продолжить свой путь до тех пор, пока тоннель не сворачивает на запад, там проход перегораживает металлическая решетка. Она, хоть и проржавела до основания, но никак не выбивается.

Итого 48%, а дальше тупик и вопросы. Как достать сухие ветки с башни, чтобы развести костер? Зачем нужны лопата и топор? Как выбить решетку? Как, если это вообще возможно, пробраться на корабль?

На остров нужно брать побольше еды, так как герои, как ни странно, хотят есть. Сэндвич можно развернуть командой UNWARP, а обертка от него, наверное, пригодится для костра. Изучайте все, за это иногда дают очки.

А теперь об игре **TERRORMOLINOS**. Эта программа понравится поклонникам HAMPSTEAD и подобных игр. Ваша цель провести выходные на испанском курорте TERRORMOLINOS, и привести обратно 10 фотографий, доказывающих, что Вы там действительно были. Первым делом, Вам нужно собрать все необходимые в путешествии вещи. Чтобы взять камеру, лежащую на кухонной полке, Вам понадобится стремянка (STEPS), но не забудьте ее застегнуть (LOCK), иначе Вам будет очень больно. Спускаться с лестницы нужно командой DESCEND. С помощью лестницы Вы должны, также, забраться на чердак, включить там свет (LIGHT ON) и, забрав все барахло, которое там пылится, покинуть это жуткое место (DESCEND STEPS). В ванной, в шкафчике, прихватите таблетки, но ни в коем случае не берите шампунь

возникнут проблемы. Из спальни захватите платок и билеты, которые вложены в брошюру. Из письма узнайте название Вашей гостиницы. Теперь, спустившись вниз, собирайте свою семью (CALL <имя>), и влезайте в такси (BOARD TAXI). Все нужно делать очень быстро, так как после 45-50 ходов Ваш самолет улетит. На этом часть, которую условно можно назвать "в доме" заканчивается.

Прибыв в **TERRORMOLINOS**, отправляйтесь в свою гостиницу. Пока Ваша жена будет заниматься багажом, не теряйте времени и приобретите фотопленку, вспышку и лосьон для загара. Вернувшись в гостиницу, можно собираться на пляж. Для этого Вам надо натереться лосьоном (RUB LOTION), одеть спортивные трусы, напялить платок, предварительно его завязав (KNOT HANK). Прихватив, также, ружье и маску, Вы можете отправляться. Не забудьте камеру с заряженной пленкой (INSERT FILM). По пути, на **PLAZA**, сделайте первую фотографию (TAKE PHOTO). На пляже, во всех локациях, кроме тех, что с водой, можно фотографировать. В море же, надев маску, можно поплавать дадут очки.

Возьмите напрокат лодку (HIRE BOAT), и плывите на остров. Там Вы увидите нечто ужасное. Чтобы избежать печальных последствий, позвоните Дорин (CALL DOREEN). Теперь Вы можете запечатлеть свою счастливую дочь. На обратном пути на Вас нападут акулы, вот тут-то и пригодится ружье (SHOOT GUN). Теперь, приладив вспышку (FIT CUBE), и взяв свою семью, сделайте фотографию в ночном клубе.

Для того, чтобы попасть в новые локации, послушайте бессвязную болтовню гида и на стоянке повозок воспользуйтесь этими командами. Для начала Вам надо в винодельню (WINE-TASTE). В подвале Вашу жену испугает паук отличный сюжет для фотографии. После этого Вы можете смело убить бедное насекомое.

Следующий пункт назначения монастырь. Мистер **SNARGSBY** тут же побежит в катакомбы, а Вы, тем временем, забирайтесь на колокольню. Какой прекрасный вид открывается отсюда! Когда Вы решите покинуть монастырь, то обнаружите, что не хватает мистера **SNARGSBY**. Что ж, придется и Вам лезть в катакомбы. Пошатавшись по диагональным направлениям, Вы найдете искомый объект. Теперь возвращайтесь.

И последний пункт бой быков. Ваша жена предложит сделать семейный портрет, наберите "Да". Дальше начнет твориться что-то кошмарное. От этих кошмаров убегайте на север, а потом на восток. Когда Вы встретите свою семью, отправляйтесь в гостиничную столовую. Там сделайте заказ (ORDER), и Вам принесут блюдо с какой-то грязью. Когда Вы съедите этот образчик местной кулинарии, Вам станет по-настоящему плохо, очень плохо. Ладно, хоть у Вас нет "Упсы" зато есть таблетки от живота.

Теперь собирайте свои вещички и отправляйтесь домой. Принимайте поздравления и попробуйте отыскать оставшиеся два процента. Может, отмыть лосьон?

Юрий Брагин из села Прелестное Белгородской обл. сообщает, как в программе **SPIDERMAN** набрать искомые 950 фунтов на печатном станке. Для этого на весы надо сбросить письменный стол, диван, **KONNERS'a** **OCTOPUS'a** и **HYDROMAN'a** в виде ледяной статуи.

Игра **SAM SPOON** трещит под натиском наших российских Шерлоков Холмсов. Очередной шаг вперед в этой игре описывает В.В.Косяченко из г. Армавир.

После того, как бандиты оставили детектива привязанным к стулу, а сами ушли по делам, у Вас есть реальный шанс убежать для этого бросьте на пол бутылку из-под бурбона (DROP BOTTLE) и пережьте осколком связывающую Вас веревку (CUT ROPE). Теперь Вы свободны!

Так же, наш корреспондент сообщает, что в этой игре, открыв что-либо командой **OPEN**, Вы потом **ОБЯЗАТЕЛЬНО** должны этот предмет закрыть (**CLOSE**), иначе Вы не сможете передвигаться, а будете все время смотреть внутрь.

Павел Козлов из г. Асбест Свердловской обл. вскрыл игру **KENTILLA**, и узнал, что найденные бриллианты (**DIAMOND**) можно надеть (**WEAR**) и, войдя в локацию с черной рекой, пройти по этой реке вниз.

Дмитрий Черников из Санкт-Петербурга полностью прошел игру **SPIDER MAN**. Он пишет, что кнопку на двери лифта нажимать не нужно, можно просто дать команду **OPEN DOOR**, а если ее нажать, то исчезает **BIO GEM** и **ENERGY EGGS**, и игру тогда уже не пройти. То же происходит, если нажать на кнопку позади вентилятора (**FAN**) или на ручку (**KNOB**) в компьютерной лаборатории. Кроме того, набрав положенные 950 фунтов на весах, надо пойти в **COMPUTER LABORATORY**, и дать команду **TYPE START PRESSES**. Вернувшись в подвал, мы найдем сложенную газету (**FOLDED NEWSPAPER**). Раскрыв газету (**OPEN NEWS**), находим последний **GEM**. Теперь идем к **MADAM WEB**, где выкладываем все **GEM'ы** и **BIO GEM**, и даем команду **SCORE**. Появится шеф-экзаменатор, который сообщит Вам код **MICHAN** и поздравит Вас с успешным окончанием игры.

Также, Дмитрий задает вопрос по игре **HEAVY ON THE MAGIC**. Он спрашивает, для чего нужно заклинание **FREEZE**. Сразу отвечаем, оно помогает Вам пройти тех врагов, убить которых Вы пока не в силах, например, вампира (если у Вас нет чеснока). После этого заклинания обитатель замка на некоторое

время заморозится (он будет подпрыгивать). В этот короткий отрезок времени Вы сможете пройти мимо него.

Олег Смолянкин из г. Киселевск Кемеровской обл. делится своими успехами в адвентюре *MAGNETIC MOON. Ему удалось сбежать с корабля. Делается это так:

В загадочном COMPARTMENT находится шлем (BUBBLE HELMET). К нему необходимо присоединить баллон с кислородом. Теперь заходим в шлюз, надеваем шлем, жмем кнопку и выходим. Сейчас самое время убраться с корабля (JUMP) благо, земля недалеко 12 метров.

Но сразу же появляются вопросы: как унести все предметы с корабля (назад уже не вернуться) и как уберечься от радиации в одной из локаций на луне?

© Константин Дзреев (Colonel), г. Ростов-на-Дону

Последние два года ZX РЕВЮ все больше и больше места уделяет играм жанра ADVENTURE. Я сам являюсь большим поклонником этого жанра благодаря циклу статей "ADVENTURE GAMES" и "ADVENTURE PROJECT".

В ZX РЕВЮ 93/9-10, 11-12 в рубрике "Сделай сам" появилась статья "Adventure Building System (ABS)". ABS является довольно неплохим инструментом для написания простеньких игр, однако алгоритм анализа предложения оставляет желать лучшего. Решив попробовать свои силы в написании adventure - игрушек, я столкнулся со следующей проблемой, которую хотелось бы вынести на обсуждение.

Каким образом корректно проанализировать введенную пользователем строку, без ошибок выделить в ней все части речи (и предложения) и правильно прореагировать на команду?

Если Вы пишете программу на английском языке, то проблем у Вас почти не будет: нет никаких окончаний, приставок и всего остального, что присутствует в нашем родном русском. У них, если в словаре программы есть слово "APPLE", то, как ни пиши "TAKE APPLE" или "THROW APPLE TO TREE", в любом случае "APPLE" останется "APPLE'ом". В нашем родном языке все гораздо сложнее. Пусть программа понимает слово "КАЛОША" (такой у нее словарь). Рассмотрим следующий пример: "ВЗЯТЬ КАЛОШУ" и "РАСКОПАТЬ КАЛОШЕЙ ПЕСОК".

Нетрудно заметить, что исходное слово из словаря отличается окончанием от приведенных в примерах.

Достаточно распространенной в англоязычных играх является идентификация слова по первым четырем (а иногда всего по трем) буквам (кстати, "ABS" работает по аналогичному принципу). В русском языке, когда только приставка в слове может занимать четыре буквы (пример - приставка "ПЕРЕ-": "ПЕРЕКИНУТЬ"), данный метод совершенно неприменим. Если производить идентификацию, минимум, по первым семи-восемью символам, то, что делать с короткими словами? Например, в выражении "КОПАТЬ ЯМУ" слово "ЯМУ" мы будем идентифицировать по всем трем буквам, а в словаре у нас слово "ЯМА", т. е. совпадения не произойдет. Альтернативным вариантом является хранение в словаре всех коротких слов во всех падежах. Но в этом случае довольно жестко встает вопрос о занимаемой словарем памяти. Вы можете отказаться ломать себе голову над этой проблемой, и перед игрой просто инструктируете пользователя о том, что программа немного не в ладах с русским языком, и все существительные нужно вводить только в именительном падеже. Приведенный выше пример будет звучать так: "ВЗЯТЬ КАЛОША" и "РАСКОПАТЬ КАЛОША ПЕСОК".

Но не думаю, что играющему в Вашу игрушку будет интересно коверкать русский язык.

Более удачным является вариант, когда в словарь программы записывается только корень слова. Наш пример в этом случае работает отлично, но и здесь не все так гладко. Отличным примером может послужить корень "ХОД": в этом случае программа одинаково прореагирует на слова "ХОД", "ХОДОК", "ХОДИТЬ", "ХОДКА" и т. п.

Но предположим, что с идентификацией слова нет никаких проблем. Тогда мы перейдем к рассмотрению целого ряда новых. Порядок слов в предложении в английском языке достаточно жестко ограничен: подлежащее, сказуемое, определение (если оно есть), дополнение. Как ни крути, по-другому не позволяет грамматика: в случае adventure игр мы вводим команду, поэтому подлежащее может опускаться: "TAKE RED APPLE" или "DIG SAND WITH STEEL KNIFE". В русском языке - полный хаос: и "АККУРАТНО ПОСТАВИТЬ КОРОВУ В СТОЙЛО", и "В СТОЙЛО АККУРАТНО ПОСТАВИТЬ КОРОВУ", и "КОРОВУ АККУРАТНО ПОСТАВИТЬ В СТОЙЛО", - все эти фразы имеют совершенно одинаковый смысл.

Достаточно легко уйти от этой проблемы, переложив ее на пользователя, т. е. подать ему очередную инструкцию, где будет написано, что ко всем прочим ограничениям программа понимает только определенно построенные грамматические конструкции, а именно: глагол + (прилагательное) + существительное

+ (предлог) + (прилагательное) + (существительное). Части речи, взятые в скобки, не обязательно должны присутствовать в предложении.

Пример: **"ВЗЯТЬ ПАЛКУ"** или **"УДАРИТЬ РЕЗИНОВУЮ УТКУ ПАЛКОЙ"**, но **"ПАЛКОЙ УДАРИТЬ РЕЗИНОВУЮ УТКУ"** программа уже не поймет, так как порядок слов в предложении не совпадает с шаблонным.

В последнем примере необходимо отметить разницу между существительными **"ПАЛКА"** и **"УТКА"**. Мы объектом **"ПАЛКА"** посредством глагола **"УДАРИТЬ"** воздействуем на объект **"УТКА"**, но не наоборот.

Далее будем называть их соответственно ПЕРВОЕ и ВТОРОЕ существительное. Обычно в программах автоматизированного создания adventure игр (пример: Graphic Adventure Creator, Quill), проверка на введенное предложение реализуется следующим образом.

Пусть глагол **"УДАРИТЬ"** имеет в словаре номер 8, а существительные **"УТКА"** и **"ПАЛКА"**, соответственно, номера 11 и 23. Тогда анализ строки может выглядеть как-то так:

If verb(8) AND noun(11) AND noun(23) THEN DO ..., что в переводе на русский язык звучит как:

Если глагол(8) И сущ(11) И сущ(23) ТО СДЕЛАТЬ ... что-то умное.

Даже невооруженным глазом видно, что два абсолютно разных по смыслу предложения (**"УДАРИТЬ ПАЛКОЙ УТКУ"** и **"УДАРИТЬ ПАЛКУ УТКОЙ"**) будут восприняты совершенно одинаково. Опять же, объемная инструкция об ограничениях программы наверняка порадует пользователя, но не слишком ли много инструкций?

Мне хотелось бы предложить некоторые соображения, при помощи которых с достаточно большой долей вероятности можно разобраться с командами типа: (наречие) + глагол + (прилагательное) + существительное + ("прямая речь"), причем порядок слов в предложении жестко не ограничивается.

Пример: **"ОСТОРОЖНО ПОСАДИТЬ ОДНОГЛАЗУЮ МУХУ НА КАМЕННУЮ СТАТУЮ"**, **"ГРОМКО ЗАКРИЧАТЬ 'ПОМОГИТЕ'"** и т. д.

Некоторые ограничения: во-первых, прямая речь должна стоять в конце предложения (ограничение легко снять), во вторых, глагол должен стоять в неопределенной форме (но это общепринято), в третьих, все слова придется вводить полностью, за исключением специально оговоренных сокращений. Некоторые достоинства:

Во-первых, порядок слов в предложении почти произволен. Программа отлично разберется и с предложением: **"НА КАМЕННУЮ СТАТУЮ ОСТОРОЖНО ПОСАДИТЬ ОДНОГЛАЗУЮ МУХУ"**.

Во-вторых, если в предложении есть предлог, то со 100% вероятностью можно будет сказать, какое из двух введенных существительных ПЕРВОЕ, а какое - ВТОРОЕ. Если предлога нет, то вероятность определения будет где-то в пределах 97 %.

В-третьих, если даже введенное слово не будет найдено в словаре, то программа определит часть речи, к которой относится неизвестное слово, и Вы сможете вывести сообщение типа: "Не знаю прилагательного **"ЦИАНОВЫЙ"** или "Впервые вижу слово **ЕЖ"**."

Алгоритм достаточно прост, и, вместе с тем, достаточно надежен. Он анализирует окончания слов. Глагол в неопределенной форме может иметь следующие окончания: **-ТЬ, -ТИ** и **-СЯ**.

Наречие всегда имеет окончание **-О** (что возможно и для существительных, и для прилагательных). Прилагательное может иметь одно из следующих окончаний: **-ЫЙ, -АЯ, -ОЕ, -УЮ, -ЫЕ, -ЫМ, -ИЙ, -ЯЯ, -ЮЮ, -ОЙ, -ЕМУ, -ОМУ, -ОГО** и т. п. (другие вряд ли будут использоваться). Особый случай окончание **-ОЙ**. Его могут иметь как прилагательные, так и существительные: **"НАСТУПИТЬ НОГОЙ НА ГОЛУБОЙ ДИСК"**. Существительные вообще могут иметь любое окончание, а отличительной особенностью предлогов является их малая длина - до 5 символов, но это не редкость и для остальных частей речи.

Шаг первый - создание пяти словарей.

С Л О В А Р И :

Глаголы	Наречия	Предлоги	Прилагательные	Существительные
1 взять	1 быстро	1 в	1 красн(ый)	1 куриц(а)
1 подобрать	2 медленно	2 на	2 прекрасн(ый)	2 песок
2 подняться	3 высоко	3 над	3 желт(ый)	3 калош(а)
3 иди	4 аккуратно	4 через	4 голуб(ой)	4 ветк(а)
4 положить	4 тщательно	5 по	5 син(ий)	4 веточк(а)
4 поместить	4 осторожно	6 под	6 красив(ый)	5 ног(а)
5 посмотреться	5 низко	6 к	7 некрасив(ый)	6 оград(а)
6 выбросить	6 точно	7 до	8 стар(ый)	6 загородк(а)
6 выкинуть	7 страшно	8 от	9 полосат(ый)	7 клоп

и т.д. и т.д. и т.д. и т.д. и т.д.

Слова, записанные в одном словаре под одинаковыми номерами являются синонимами. У прилагательных и существительных опускаем окончание, то есть, оставляем приставку, корень и суффикс.

Шаг второй - написание программы, корректно анализирующей введенную пользователем строку, и также корректно ее исполняющей. На этом шаге остановимся.

Я не буду очень подробно приводить алгоритм, а только кратко опишу его работу. Подробно описанный на языке высокого уровня алгоритм, достаточно велик по объему.

Суть алгоритма заключается в следующем. Пусть пользователь ввел какую-либо символьную строку. Для начала надо отделить в ней команду от прямой речи (многие наверняка знают, что это такое).

Делается это следующим образом: находим в этой строке символ-разделитель прямой речи и команды - обычно это одинарная или двойная кавычка. Все что слева от кавычки - это команда, все что справа - это прямая речь.

Например: **"СКАЗАТЬ ДЖИНУ 'ЛЕЗЬ В БУТЫЛКУ'"**. Командой является **"СКАЗАТЬ ДЖИНУ"**, а прямой речью **"'ЛЕЗЬ В БУТЫЛКУ'"**. Будем анализировать команду.

Во-первых, в строке необходимо найти глагол. Поиск производим следующим образом: ищем слово с окончанием : **-ТЬ, -ТИ, -СЯ**. Если слово найдено, то сравниваем его со словами из словаря глаголов. Если такого слова программа не знает, то ищем в предложении другое слово с таким окончанием. Это делается из-за того, что вышеперечисленные окончания могут иметь не только глаголы, но и существительные: **"СХВАТИТЬ ГУСЯ ЗА ХВОСТ"**. Если предложение просмотрено полностью, а известных программе слов нет, то самое время высветить сообщение об ошибке: "Не вижу здесь глагола!"

Сравнение слов необходимо производить следующим образом: на слово ИЗ СЛОВАРЯ накладываем слово ИЗ ПРЕДЛОЖЕНИЯ, но не наоборот. Пусть в словаре есть слово **"ПОСМОТРЕТЬСЯ"**. Тогда, накладывая на него любое из слов **"ОСМОТРЕТЬСЯ", "ОСМОТРЕТЬ", "ПОСМОТРЕТЬ"**, мы получим совпадение, то есть не надо держать в словаре эти три слова, как синонимы, а достаточно только одного. Если глагол найден, то необходимо убрать его из предложения, чтобы он не мешал дальнейшему анализу. Далее будем изымать из предложения каждое из успешно идентифицированных слов.

Далее, в оставшемся предложении попытаемся отыскать наречие. Как известно, наречие всегда оканчивается на **"-О"**, а по сему, проделываем для него все вышесказанное для глагола, но только поиск ведем в словаре наречий, и, в случае, если наречия не окажется или оно не будет опознано (т. е. слово с окончанием **"-О"** есть, но программа не знает его, как наречие - может быть это слово **"ЯБЛОКО"**), сообщения об ошибке не выдаем.

Затем найдем предлог. Предлог ищется по принципу: если длина слова меньше шести букв, то возможно это слово и есть предлог.

Как только короткое слово найдено, оно тут же пропускается через словарь предлогов, и, если такого предлога в словаре нет, то искать и анализировать надо следующее короткое слово. Отсутствие предлога не смертельно для предложения. Поэтому никаких сообщений о несовпадении или об отсутствии этой жизненно важной части предложения нет.

Продолжим и будем искать прилагательное. Прилагательных в предложении, как и существительных, может быть два, а может не быть вообще. Кроме того, после любого прилагательного обязательно должно стоять существительное, к которому это прилагательное относится (**"КИНУТЬ В ПРОЗРАЧНУЮ ВОДУ БЕДНУЮ МУ-МУ"**). Прилагательные распознаются по своему окончанию : **-ЫЙ, -АЯ, -ОЕ, -УЮ, -ЫЕ, -ЫМ, -ИЙ, -ЯЯ, -ЮЮ, -ОЙ, -ОМУ, -ОГО**. Пусть программа обнаружила слово с любым из приведенных окончаний. Тогда она обрезает это окончание и сравнивает полученное слово со словарем прилагательных. Если слово не распознано и окончание его на **"-ОЙ"** (такое окончание может иметь существительное), то выдать сообщение об ошибке : "Впервые вижу прилагательное БЕДНУЮ". В случае отсутствия прилагательного в предложении - ошибки нет.

И, наконец, существительные. Их идентификация не является особо сложной задачей. Проблема может возникнуть при анализе того, какое из них ПЕРВОЕ, а какое ВТОРОЕ. Если в предложении был предлог, то определенно можно сказать, что стоящее после предлога существительное - ВТОРОЕ, а оставшееся - наверняка ПЕРВОЕ. Но, ЕСЛИ ПРЕДЛОГА НЕТ, то возникают проблемы. Рассмотрим несколько примеров. **"РАСКОПАТЬ ПЕСОК КАЛОШЕЙ", "УДАРИТЬ ВЕТКОЙ КУРИЦУ", "ЗАДАВИТЬ КЛОПА НОГАЙ"**.

А теперь сравните все существительные с их аналогами в словаре (см. выше). Не заметили? ПЕРВЫЕ существительные имеют окончания из двух букв, ВТОРЫЕ - либо вообще не имеют, либо окончание состоит всего из одной буквы. Таким образом, наложив слово из словаря на слово из предложения и проанализировав оставшееся окончание (по числу символов), можно определенно сказать кто чем на кого действует. Если в предложении всего одно существительное, то вышеописанных проблем вообще не возникает.

Анализ прямой речи можно свести к анализу простой команды, но необходимо учитывать, что глагол, скорее всего, будет стоять не в неопределенной форме.

Более "умные" алгоритмы при идентификации слова распознают в нем и приставку, и корень, и суффикс, и окончание. Но в этом случае процедура анализа предложения может занимать десяток-другой килобайт, что в случае Sinclair - совместимых компьютеров абсолютно неприемлемо (разве что у Вас 256-й Scorpion или ATM-TURBO 512-й). Но такой сложный алгоритм просто не нужен, ведь все разнообразие диалоговых ситуаций в хорошей ADVENTURE игре можно реализовать и при помощи более простых алгоритмов.

СОВЕТЫ ЭКСПЕРТОВ



ROCK STAR ATE MY HAMSTER

Эксперт: Олег Бобрышев, г. Георгиевск.

Жанр: Business/Management

В этой игре Вам предстоит попробовать свои силы в музыкальном бизнесе. Начинается игра с того, что Ваше театральное агентство разоряется. После этого между Вами (боссом) и Вашим помощником Клайвом (исполнительным продюсером) происходит следующий диалог:

Босс: Мы прогорели! Даже Элвин Маниллоу, танцующий жираф, бросил нас! Все, что у нас есть это пятьдесят тысяч, которые прислала тетушка Мейбл. Никогда больше не свяжусь с театром! Настоящие деньги только в рок-бизнесе! (Эта, и дальнейшие реплики могут варьироваться в деталях. Видимо, они выбираются программой случайно.)

Клайв: Ух! Рок-н-ролл! Пройдемте же в просмотровый зал...

В дальнейшем все запросы программы идут в виде вопросов Клайва. Судя по ним, Клайв исполнительный, боготворяющий начальство, продюсер, но начисто лишенный инициативы (а может, мозгов?).

В первую очередь Клайв спрашивает, сколько человек Вы хотите иметь в составе группы. Чем меньше человек, тем легче с ними иметь дело, но, с другой стороны, чем их больше, тем устойчивее группа и тем больше шансов выжить, если Вы потеряете кого-нибудь. (Об этом ниже). Кроме того, от состава Вашей группы будет напрямую зависеть качество песен, которые они будут исполнять. Если набирать коллектив наобум, последствия (в виде качества звука) могут быть ужасными. Вместо мелодии будет звучать какофония из отдельных звуков, непереносимая на слух. Правда, я не заметил, сильно ли это влияет на объем продаж, но кажется, что зависимость есть. Кстати, узнать "любимый" стиль исполнения каждой звезды можно, выбирая их по одному и прослушивая.

После того, как Вы определились с количеством звезд, Вам необходимо выбрать, кого же именно Вы хотите видеть в своей группе. Список исполнителей очень обширен 50 человек, так что есть из кого выбирать. Большинство (если не все) из предлагаемых Вам кандидатов это шаржи на реальных исполнителей. Так, например, WACKY JACKO это, несомненно, Майкл Джексон, а в девушке по имени MARODONNA легко узнается Мадонна.

Цена звезды (плата в неделю) напрямую не связана с ее популярностью и "двухсотфунтовая" звезда может собирать полные залы, тогда как исполнитель за 20000 фунтов будет все еще "раскручиваться". Но подмечено, что, как правило, чем больше Вы отдаете за звезду, тем меньше будет стоить ее оборудование. Так что, в подборе руководствуйтесь исключительно личным вкусом и финансовыми возможностями.

После того, как Вы выберете своих звезд, Клайв предложит Вам как-нибудь назвать получившуюся группу. При этом можно согласиться с тем названием, которое Клайв Вам предлагает, а можно и самому назвать эту группу.

Когда группа названа, необходимо выбрать ей оборудование музыкальные инструменты, звукоусиливающую аппаратуру и пр. Вы можете выбрать оборудование трех категорий:

Абсолютно новое (SPANKING NEW)

Бывшее в употреблении (SECOND HAND)

"Со свалки" (DODGY GEAR).

Цены здесь будут зависеть, как я уже говорил выше, от того, каких исполнителей Вы выбрали.

Наконец, после выполнения предварительной подготовки, Вы попадаете в основное меню. Оно состоит из следующих пунктов:

1. PRACTICE - репетиция.

2. GIG - концерт.
3. PUBLICITY общественное мнение.
4. GIFTS подарки.

Остановимся на каждом подробнее:

1. PRACTICE. Перед началом Вашей концертной деятельности группе необходимо попрактиковаться. Иначе она будет повышать уровень игры на сцене, что отнюдь не улучшит посещаемость концертов. Как правило, любой группе требуется шесть дней практики. Если же у Вас один исполнитель, то это время сокращается.

После выбора пункта PRACTICE, Клайв сообщает, что репетиционное помещение стоит 100 фунтов в день и предлагает выбрать количество дней (от одного до пяти). Здесь Вы (как игрок) можете послушать, как будет играть Ваша группа или (как босс) поспать, пока парни заняты своим делом. Клайву, естественно, нравится все, что Вы делаете, а также все, что играют звезды. Поэтому по окончании практики Вы услышите от него нечто похожее на следующее: "Невероятно, босс! Бьюсь об заклад, так чудесно, как наша группа, никто не играет!"

2. GIG - После практики Вам следует заняться концертами. Это основной для группы источник денег, а кроме того, только на концертах Вас могут увидеть Ваши будущие спонсоры. После выбора этого пункта, Вам будет представлено дополнительное меню. Вам нужно оговорить конкретные условия проведения концерта. Это:

2.1. Место проведения. Можно выбрать между:

2.1.1. PUBS бары. 200 мест, 10 фунтов за ночь (все рок-концерты проходят ночью).

2.1.2. CLUBS клубы. 500 мест, стоимость помещения 1000 фунтов за ночь.

2.1.3. UNIVERSITIES университеты. 1500 мест, 2800 фунтов за ночь.

2.1.4. CONCERT HALLS концертные залы. 4000 мест, 7000 фунтов за ночь.

2.1.5. STADIUMS стадионы. 10000 мест, 16500 фунтов за ночь.

Когда Вы выберете, где будет петь Ваша группа, Вам придется определить следующее:

2.2. Цена билета: Может быть 2, 5, 10, 15 и 20 фунтов. От размеров цены зависит, сколько человек придет к Вам на концерт, а значит сколько Вы заработаете. Кроме того, цена определяет и популярность как много людей придет на следующий Ваш концерт. Если цена билета мала, здание, где Вы выступаете, забьют до отказа уже на второй день, но денег с этого Вы будете иметь немного. Наоборот, при цене билета в 20 фунтов, людей придет не так уж много, зато на каждый следующий концерт на 100 человек больше. Сколько денег это принесет!

Следует заметить, что приведенные выше соотношения действуют только на начальном этапе. Когда у Вас будет достаточно денег, чтобы выступить в концертных залах и на стадионах, Вы обнаружите, что происходит следующее: концертные залы могут полностью наполняться при цене билетов до 15 фунтов. Если же поставить 20 фунтов за билет, то заполнится лишь три четверти зрительских мест. На стадионах и того хуже все билеты раскупаются по цене 10 фунтов и ниже (что дает максимальную прибыль $10 \times 10000 = 100000$ фунтов). При цене 15 фунтов за место, заполняется около двух третей стадиона ($15 \times 6000 = 90000$ фунтов), а по 20 фунтов не продашь и половины билетов ($20 \times 4000 = 80000$ фунтов). Таким образом, оптимальная стратегия на стадионе продавать билеты по 10 фунтов.

Наконец, Вам придется определить, как долго Вы будете занимать выбранное помещение:

2.3. Время гастролей. Варьируется от двух до семи ночей. Определяется Вашими финансовыми возможностями (платить за наем Вам придется сразу, из наличных денег), а также разумными ограничениями (глупо выступать в переполненных клубах, когда полупустые университеты дадут гораздо больше).

В любой момент Вы можете отказаться от гастролей, выбрав пункт CANCEL в меню.

Ну вот, Вы определились с графиком гастролей и они, наконец, начинаются. Ваше в них участие ограничивается в основном руководящими указаниями типа: "Ну, ребята, идите и убедите их всех своей музыкой!" Кроме того, здесь Вы узнаете (от Клайва, естественно), как разошлись билеты. Если объем продаж падает, лучше сходить и потренироваться еще. Кстати, падает он в той же пропорции, что и растет, то есть при цене билета, скажем, в 5 фунтов зал мгновенно наполняется, но и опустошается тоже мгновенно. Если же билет стоит 20 фунтов, зал медленно наполняется (на сто человек больше, чем на предыдущем концерте), но и теряете Вы зрителей теми же темпами по сто человек за концерт.

Кроме того, во время гастролей могут возникнуть трения со звездами. Выглядит это примерно так: какая-нибудь из звезд заявит Вам следующее: "Я считаю, что гигантский фейерверк значительно улучшит наше представление. Он стоит всего 70000 фунтов. Ты согласен???"

Если Вы согласитесь с такими условиями, ничего не произойдет (кроме того, что из Вашего бюджета исчезнет 70000). Но если Вы откажете, ситуация имеет продолжение. Возможны два варианта:

Первый вариант. На следующий день Клайв говорит: "Босс, у нас тут прошлой ночью возникли небольшие проблемы..." После этого Вам покажут передовицу газеты "THE STUN REPORTER" с заголовком типа "Рок-звезда разгромила комнату в отеле!", после чего Клайв поясняет, что выплаты составили, например, 40000 фунтов. Эти деньги автоматически вычитаются из Вашей наличности. На этом все и заканчивается.

Второй вариант. На следующий день та же рок-звезда снова приходит к Вам. Теперь она уже требует в следующих выражениях: "Немедленно предоставьте мне сценический костюм, украшенный бриллиантом за 90000 фунтов, или я уволюсь!" Здесь у Вас есть выбор. Вы можете выполнить требования, и все опять-таки на этом кончится. Но Вы можете и отказать. Тогда звезда получит от Вас следующее заявление: "Ну, ты, напыщенное вонючее ничтожество! Ты не уволишься, я тебя выбрасываю!!!" После этого в группе на одного участника станет меньше...

Такая потеря звезды сильно влияет на сборы количество людей на следующем концерте будет меньше на треть.

Теперь рассмотрим третий пункт меню общественное мнение.

3. PUBLICITY. Если Вы выберете этот пункт, то услышите от Клайва примерно следующее: "Оставьте это мне, босс! Уж я-то знаю, как делаются такие вещи!" Дальнейшие результаты Вы увидите в газете "THE STUN REPORTER". Они могут быть трех видов:

3.1. Успех. Заглавное сообщение газеты гласит что-нибудь типа "На рок-звезду напали пчелы-убийцы!". Играет (предположительно) радостная мелодия. Клайв заявляет: "Ну, как у меня получилось?!"

3.2. Неудача. На обложке позирует полуголая красотка. Заголовок гласит, например: "Викарий поженит инопланетянина!" Звучит (опять-таки предположительно я не вижу между ними большой разницы) грустная мелодия. Клайв заявляет: "Ну и что, босс?! Жизнь есть жизнь, разве не так?"

3.3. Смерть звезды. На обложке заголовок: "Рок-звезда погибла в атомной войне!" Несмотря на это, мелодия звучит радостная всё же реклама группе. Клайв говорит: "Не расстраивайтесь босс! Она отправилась в великий дворец звукозаписи туда, на небеса!" Вы лишились одной звезды, и посещение падает, как если бы Вы ее уволили.

В случае успеха публикации Ваши записи будут лучше продаваться (об этом ниже). В случае неудачи, видимо, результат нулевой только день потеряли.

Наконец, последний пункт меню:

4. GIFTS. Как я уже писал выше, во время гастролей могут возникнуть конфликты с музыкантами. Для их предотвращения и служат подарки.

При выборе этого пункта Клайв говорит: "Великолепно, босс (это его дежурная фраза). Хотите сохранить наших ребят счастливыми, да? Что же мы дадим нашим изумительным рок-звездам? Цены на одну звезду следующие..." Далее следует список подарков:

4.1. CARS (Машины). 20000 фунтов.

4.2. JEWELS (Драгоценности). 10000 фунтов.

4.3. CLOTHERS (Одежда). 5000 фунтов.

4.4. DRINKS (Напитки). 100 фунтов.

4.5. FOODS (Еда). 50 фунтов.

4.6. COMICS (Комиксы). 1 фунт.

В этом списке пункты с первого до четвертого (от машины до выпивки) примерно равнозначны и действуют от недели до месяца в зависимости от того, насколько они нужны сейчас группе. При выборе конкретного подарка следует ориентироваться на сообщения, выдаваемые звездами. Пункты же 5 и 6 могут применяться лишь как временное средство, если уж совсем нет денег они помогают дня на два, на три; максимум на неделю.

Следует заметить, что цены указаны в расчете на одну звезду. То есть, если в составе Вашей группы три человека, и Вы покупаете одежду, то заплатите 15000 фунтов по 5000 на каждого.

Ну а теперь, когда Вы ознакомились с основным меню, собственно игра. Вам следует, сначала попрактиковавшись, набрать достаточную сумму денег, чтобы жить, не опасаясь за завтрашний день. Помните, что каждое действие (естественно, кроме тех, где время указано явно) займет у Вас один день, а каждую пятницу Вам необходимо платить жалованье.

По мере того, как будет расти Ваша известность, к Вам будут обращаться с различными звонками. О большинстве из них я сообщу ниже, а сейчас о самых важных звонках от компаний звукозаписи. Чем выше Ваша популярность, тем более крупные компании к Вам обращаются. Самой первой звонит FLOPPY DISCS RECORDS. Она предлагает 5000 фунтов и 3% от объема продаж. Вообще-то, не советую соглашаться на эти условия. Если Вы откажетесь, то по мере заполнения помещений (от университетов к концертным залам, а от них к стадионам), к Вам будут обращаться все более крупные компании. Наконец, после запол-

нения стадиона, следует звонок от MEGA BIG RECORDS. Они предлагают 50000 фунтов и 20% от объема продаж. Тут уже отказываться ни в коем случае нельзя больше звонков не будет! Конечно, Вы можете выступать, делать деньги, но к конечной цели не приблизитесь ни на шаг.

Ну и, наконец, о конечной цели игры. Она состоит в том, чтобы выпустить четыре "золотых" диска (или, что-то же самое, один "платиновый"). Диски изображены в левой нижней части экрана. Как только Вы сделаете очередной "золотой", он загорится. Определяется, стал ли Ваш диск "золотым" или нет, по объему продаж. Колебания объема продаж отражает график справа. Чем больше черная часть, ограниченная графиком, тем ближе Вы к очередному "золоту". Между дисками и графиком находится календарь. По нему Вы можете определить, далеко ли до следующей выплаты звездам, сколько осталось до хит-парада, да и вообще, сколько времени Вам еще играть.

Выплаты звездам проходят каждую пятницу, хит-парады по воскресеньям. Для того, чтобы выполнить поставленную задачу, Вам дается один год с 1 января по 31 декабря.

Ладно, вернемся к нашим баранам, то есть к компаниям звукозаписи. После того, как Вы согласитесь на условия любой из них, в меню появится еще один пункт:

5. RECORD (Запись). Не стоит сразу же спешить воспользоваться им, если у Вас не очень много денег (меньше полумиллиона). Но и особо тянуть тоже не стоит чем раньше Вы запишитесь, тем раньше выйдут Ваши пластинки и тем раньше они распродадутся. Если Вы вошли в этот пункт, перед Вами появится следующее меню:

Какое оборудование для записи Вы выбираете?

- Запись на 2 дорожки за 2000
- 4 дорожки за 5000
- 8 дорожек за 10000
- 16 дорожек за 16000
- 24 дорожки за 20000
- 48 дорожек за 50000
- CANCEL (отменить выбор)

Следует отметить, что время записи напрямую зависит от студийного оборудования. При записи на 48 дорожек на одну песню уходит один день, при записи по 24 два дня, по 16 три дня и т.д. Поскольку во время записи деньги не поступают, естественно, лучше брать наиболее "быстрое" оборудование. (Вот Вам причина, почему лучше подождать с записью).

После того, как Вы разобрались с оборудованием, начинается собственно запись. Она состоит в том, что Вам проигрывают 10 мелодий и просят дать им название. Можно оставлять и те названия, которые предлагает Клайв. Чтобы Вы не очень скучали (точнее, когда Вы, как босс, заскучали) изредка происходит следующий диалог:

Босс: А что вот это за кнопочка? Можно ее нажать?

Клайв: Стойте! Поздно... Шеф, Вы только что стерли нашу последнюю запись!

После этого запись последней песни повторяется. Хорошо, если Вы писали на 48 дорожек потеряете день. А если на 24? Или на 16?! Очень обидно...

Наконец, альбом записан. Но это еще не все! Если у Вас, по какой-то случайности после записи осталось больше миллиона, всегда раздастся телефонный звонок и Клайв сообщает: "Шеф, продюсер этой студии удрал вместе с нашим альбомом и требует миллион! Мы ему заплатим?" Если Вы платите, то остаетесь с альбомом, но без миллиона, а если нет, пишете альбом по второму кругу. (Вот Вам причина, чтобы не тянуть с записью).

Но вот, у Вас на руках записанный альбом. Теперь пункт RECORD заменяется другим:

- 6. RELEASE (Выпустить). При выборе этого пункта появляется подменю:
 - 6.1. Выпустить альбом.
 - 6.2. Выпустить сингл (одну песню на диске).

Первый пункт всегда будет гореть, пока Вы не выпустите альбом. После выпуска альбома подменю не выводится программа автоматически считает, что Вам нужен п.2

Итак, 6.1. Выпустить альбом. Этим пунктом лучше не пользоваться, пока у Вас нет хороших видеоклипов. А именно, если Ваш видеоклип не попал в лучшую десятку или даже восьмерку клипов, альбом падает на одну позицию каждую неделю.

Если Ваш клип занимает места от 8-го до 6-го, то альбом будет подниматься лишь до первой пятерки. Если же альбом уже был в пятерке лучших, он начнет опускаться на одну позицию каждую неделю.

Если видеоклип занимает место с 5-го по 2-е, альбом будет подниматься по одной позиции за неделю. Ну, а если Ваш клип поднялся на первое место, то популярность альбома будет увеличиваться ударными темпами по пять позиций в неделю!

При этом следует отметить, что, выйдя из лучшей десятки, Ваш альбом будет продолжать падать, и его перемещения запоминаются, но уже не отражаются в хит-параде. И для возвращения в "TOP 10" ему потребуется вновь пройти все утерянные позиции. Быстро вернуть альбом в хит-парад можно только в том случае, если Ваш клип займет первое место.

Выбрав пункт 6.1. Вы должны будете дать название новому альбому. На этом Ваши обязанности по отношению к нему и заканчиваются.

6.2. Выпустить сингл. Здесь Вам предоставляется обширный выбор. Сначала Клайв спрашивает Вас, какой из десяти Ваших хитов Вы хотите выпустить. Если название песни длиннее одной строчки, его сократят до последнего слова, но только в этом пункте в хит-параде название будет присутствовать полностью. Так как пробел или точку программа считает словом, не удивляйтесь, увидев вместо названия пустое место или знак препинания лучше вспомните, что Вы записывали.

Затем Вас спросят, хотите ли Вы делать видеоклип, или просто выпустить песню. Если Вы не будете делать видеоклип, Ваши затраты на этом кончатся. Правда, и результат будет соответствующим я так ни разу и не увидел таких песен в десятке.

Но если Вы делаете видеоклип, у Вас появится множество забот об его организации. Во-первых, Вам нужно выбрать режиссера клипа от Стивена Чизбургера (STEVEN CHISBURGER) за 3000000 фунтов до Клайва (CLIVE) за 5 фунтов. Сами понимаете, как будет управлять Клайв, и как известнейший режиссер, фамилия которого явно пародирует Стивена Спилберга. Я считаю, нужно брать самого дорогого режиссера, которого Вы можете нанять.

Во-вторых, необходимо выбрать место съемок. Возможные варианты:

BAHAMAS (Багамы) 50000 фунтов.

HOLLYWOOD (Голливуд) 20000 фунтов.

PINEWOOD (Пайнвуд, дословно сосновый лес) 10000 фунтов

CRICKLEWOOD (Криклвуд, название можно перевести как Сгоревший Лес) 50 фунтов.

Последний пункт этого меню явное издевательство (сравните цены!) Относительно же Багамских островов и Голливуда можно сказать, что они приблизительно равноценны весь вопрос, что снимать. Натурные съемки, естественно лучше проводить на Багамах, а игровые в Голливуде.

После выбора места необходимо выбрать сюжет видеоклипа. Вам дается выбор из шести пунктов, которые могут сильно варьироваться от клипа к клипу. Возможно, они зависят от самой песни, возможно, от места съемки или режиссеров, а может быть от всего сразу. Вот типичный пример:

LUSTY GIRLIES MUD WRESTLING (Борьба в грязи со страстными девушками).

TOTALLY TACKY PUPPETS (Полностью липкие марионетки).

GREASY TOY BOYS PUMPING IRON (Потные игрушечные парни, "качающие железо").

SICK CAR CHASES (Гонки до тошноты).

VAMPIRES, SPIDERS & RATS (Вампиры, пауки и крысы).

MIND-NUMBING GUITAR POSES (Ошеломительные гитарные сцены).

Ваш выбор может определяться лишь личными пристрастиями.

Но вот, Вы выбрали и действующих лиц. Что делать теперь? Ответ ничего! Точнее, отправляться на гастроли до ближайшего воскресенья. В воскресенье Вам покажут результаты последнего хит-парада, и Вы поймете, стоит ли выпускать альбом, или еще подождать. Если же альбом уже выпущен, можно подумать и о следующем клипе.

По поводу видеоклипов есть несколько советов:

- Если клип не "выплыл" за пять недель, лучше выпустить новый старый уже не появится в десятке, а если и появится, то не поднимется выше 7-го, 8-го места.

- Если видеоклип упал ниже 7-го места, выпускайте новый, старый больше не поднимется.

- Общественное мнение лучше формировать, пока популярность клипа растет. Когда она начнет падать, публикации лишь задержит его в десятке чуть дольше.

- Никогда не выпускайте свежих видеоклипов, если старые еще достаточно высоко. Дело в том, что одновременно в хит-параде может быть только один Ваш клип, а именно последний выпущенный. Вы просто потеряете деньги, которые могли бы заработать на предыдущем видеоклипе.

Все, сказанное выше, относится только к Вашей группе. У Ваших же конкурентов не наблюдается никакой связи между популярностью альбома и видеоклипов.

И еще одна область, которую я почти не затронул это телефонные звонки. Их можно разделить на несколько групп:

1. Звонки от компаний звукозаписи. О них я говорил выше.

2. Звонки от спонсоров. Поскольку трубку всегда берет Клайв, звучит это примерно так: "Эй, шеф! Тут мужик на телефоне, говорит что из "Ливиес Джинс" (LEAVIE'S JEANS). Он хочет стать нашим спонсором и предлагает 50000 фунтов. Согласиться?"

Соглашаться не следует никогда! Дело в том, что Вы приобретете:

а). 50000 фунтов.

б). Имя спонсора (вместо Вашего) на афише.

И все! Зато Вы потеряете зрителей, и весьма существенно: на Ваш концерт станет приходить на 800-900 человек меньше, чем до этого. Вам это, естественно, ни к чему.

3. Звонки от различных благотворительных обществ. Делятся на два типа: Звонки от тех, кто хочет Вас послушать, и от тех, кто хочет над Вами посмеяться. На начальном этапе Вы никак не можете отличить одних от других. Результаты же следующие:

а). Звонки от любителей музыки. Если Вы отвечаете "да", играет радостная (все та же) мелодия и у Вас увеличивается популярность. Если же ответ "нет" в "STUNT REPORTER" появляется статья, под заголовком "Рок-звезда грубо отказала в проведении концерта". Популярности это никак не способствует, сборы уменьшаются вдвое.

б). Звонки с целью посмеяться. Здесь ситуация обратная. Если Вы отвечаете "нет", играет радостная музыка и все остается, как было. Если же Вы согласитесь дать благотворительный концерт, появится статья в "STUNT REPORTER", озаглавленная: "Рок-звезду освистали из-за излишнего милосердия". Как Вы понимаете, лучше Вам от этого никак не станет результаты, что и в предыдущем случае.

Как я уже сказал выше, по внешнему виду с первого взгляда эти звонки никак не различаются. Однако я заметил, что Клайв произносит одинаковые фразы, когда звонят одни и те же фирмы. Вернее, совпадают начало и концовка реплики Клайва, то есть высказывание, начинающееся с HEY CHIEF!, а заканчивающееся WE ACCEPT? соответствует звонку одной организации, а LISTEN BOSS ... WE SHOULD? другой. И если фирма позвонит Вам в следующий раз, Вы будете знать, что отвечать.

4. Звонки о пиратстве. Клайв реагирует на них так:

"Слушайте, шеф! Маленькая компания в Кенсингтоне копирует наши диски! Что мы будем с этим делать?"

Далее выводится меню из четырех пунктов:

DO NOTHING ничего не делать.

BUY THE ... купить за ...

SEND BOYS AROUND послать ребят разобраться.

SUE THE SCUM засудить мерзавцев.

Выбор пункта целиком определяется местонахождением пирата. Если ничего не сделать, упадет объем продаж. Если же выбрать что-то другое, следует помнить, что в Корею накладно посылать ребят "на разборки" это слишком далеко. А вот с пиратами в Кенсингтоне лучше не церемониться в суде, иначе процесс может затянуться.

И напоследок, об уровнях игры. Дело в том, что не все так просто, как кажется на первый взгляд. Чтобы Вам не стало совсем уж скучно, после каждого "золотого" диска происходит усложнение игры:

После первого "золота" у Вас портится оборудование. Клайв сообщает Вам: "Все наше оборудование превратилось в пыль! Хотите купить новое?" После этого Вы выбираете оборудование, как и вначале. Кроме того, у артистов резко повышается самомнение и они начинают часто капризничать.

После второго "золота" Вас ждет судебный иск. Клайв говорит: "Боб, Майк и Билли заявили, что мы откопали их старую песню для группы BAZOOMA ROOMA! В суде мы потеряли миллион!". И одним миллионом у Вас становится меньше. Чаще звезды уже не капризничают, зато они повышают требования. Теперь они увольняются сразу же после первого отказа, а не приходят на следующий день выдвигать условия.

После третьего "золотого" диска вновь проблемы. Клайв: "У нас возникли проблемы с транспортом. Накладные расходы составили три миллиона!" Если у Вас нет трех миллионов, то здесь с Вами и распродаются, но даже если и есть, не спешите радоваться. Дело в том, что теперь звезды окончательно поверили в свою гениальность и увольняются при первом же возникшем у них требовании. Вам остается либо заваливать их подарками каждую неделю, либо смириться с риском получить несколько раз подряд сообщение "Звезда уволилась!" и закончить игру. Именно так несколько раз случалось и со мной...

Но, если Вы выжили и после этого, и до конца года еще есть время, у Вас есть шанс заработать четвертый "золотой" диск и стать обладателем "платины"! Однажды мне это удалось, Сообщаю результат: программа останавливается (чем бы Вы не занимались, Вы оказываетесь в офисе). Клайв радостно сообщает:

- Ура, босс! Мы только что стали обладателями "платинового" диска!

Вам, как боссу, он видимо изрядно надоел за этот год, поэтому в ответ на его реплику Вы говорите:

- В самом деле? Клайв, ты уволен! Уж теперь я смогу, наконец, найти продюссера лучше тебя!

После этого Вам будет предоставлена возможность сыграть заново с этими звездами, или выбрать новых.

Для 128-к версии, видимо, предусмотрена дополнительная музыка, а может и еще что-нибудь. Не могу сказать чего именно, такой версии у меня нет, но в загрузчике для 128-го компьютера предусмотрена подгрузка еще двух блоков.

DUNDEES EUROPEAN CHALLENGE

Автор: Wayne Easson

Эксперт: Дмитрий Сухарев, г. Тольятти.

Жанр: Business/Management

В этой игре Вы становитесь одним из девятнадцати игроков и ассистентом тренера шотландской команды Dundee. Загрузив игру, Вы увидите вопрос: LOAD A SAVED GAME? Y OR N

Нажав "Y", Вы сможете загрузить игру с внешнего носителя. Если же Вы нажали "N", то у Вас спросят Ваше имя:

TYPE IN YOUR NAME. MAX 16 CHARACTERS (Введите свое имя, максимум 16 символов).

После этого появится состав команды из 19 фамилий, последняя из них Ваша. В правой части экрана изображена вертикальная полоска. Это европейский рейтинг Dundee. Нажав любую клавишу, Вы переходите к основному экрану игры тотализатору.

Здесь Вы увидите три окна:

Верхнее окно: в этом окне показаны предстоящие или прошедшие матчи (в последнем случае будет показан счет).

Левое нижнее окно: Здесь Вы можете указать исходы четырех матчей, так как команд всего восемь.

- Нажав "H", Вы предсказываете победу команды, играющей дома.

- Нажав "D", Вы предсказываете ничью.

- Нажав "A", Вы предсказываете победу команды, играющей на выезде.

После матча, напротив результатов, указанных Вами, появятся знаки:

V результат матча указан верно.

X результат матча указан неверно.

За три правильно угаданных исхода Вам дадут 25000 фунтов, а за четыре 50000 фунтов.

Правое нижнее окно: таблица лиги. Здесь Вы увидите, какое место заняла каждая из команд и количество набранных ею очков (PTS). За выигрыш дается 2 очка, за ничью 1, а за поражение 0 очков.

Если у Вас после матча будет 35000 фунтов, то Вас спросят:

TRANSFER MARKET? Y/N (Хотите ли Вы обменять какого-либо игрока).

При положительном ответе появятся 18 фотографий игроков, которых Вы можете обменять. Под изображением игрока находится его фамилия и стоимость обмена.

Для того, чтобы обменять игрока, нужно набрать его номер и нажать ENTER. После этого появится имя игрока, которого Вы заменили, а затем новый состав и рейтинг команды.

ИГРА

Игра начинается после того, как Вы указали исходы всех матчей в тотализаторе. В процессе матча Вам будут показывать наиболее интересные моменты встречи.

Вверху расположены два окна: Левое окно табло. Здесь изображаются названия команд и счет матча. Правое окно информационное, в нем возникают сообщения, отражающие события на поле:

What a Goal! – «гол!»

That was close – мяч прошел рядом с воротами.

What a save! – «вратарь отбил!»

Crowf – количество зрителей, пришедших на матч.

Receipts – количество денег, вырученных от продажи билетов.

Еще сообщения выводятся на игровом экране:

Penalty. ref – пенальти

Foul ref. freekick – свободный удар.

Если Ваша команда забьет гол, Вы узнаете имя отличившегося игрока.

После окончания матча появится изображение тренера, который скажет Вам все, что о Вас думает.

SPACE – дальше;

S записать состояние игры на внешний носитель.

В конце сезона Вы будете бороться за Европейский кубок, где будут участвовать такие сильные команды, как "Аякс", "Ювентус" и т.д. Чтобы европейский рейтинг Вашей команды быстрее возростал, постарайтесь при первой же возможности менять игроков.

Удачного чемпионата!

WINTER GAMES

Фирма: EРУХ

Эксперт: Андрей Рубин, г.Энгельс, Саратовской обл.

Фирма EРУХ подарила Вам уникальный шанс поучаствовать в шести различных видах зимнего спорта! Прекрасная графика, безусловно, не оставит Вас равнодушным к этой игре. Ну, а если Вы еще и любитель зимних видов спорта - эта игра для вас.

Итак, Вы загрузили WINTER GAMES. На экране появится меню:

1. Все виды соревнований.
2. Виды соревнований по выбору.
3. Участие в одном соревновании.
4. Тренировка.
5. Выбор количества игроков.
6. Выбор управления.
7. Таблица мировых рекордов.

Рассмотрим все пункты по порядку:

1. Вам предлагается возможность участвовать в шести видах спорта, один за другим:

1. HOT DOG
2. BIATHLON
3. SPEED SKATING
4. SKI JUMP
5. FREE SKATING
6. BOBSLEY

HOT DOG - одно из упражнений фристайла. Вам нужно выполнить это упражнение в прыжке с трамплина. Клавиши управления прыгуном:

FIRE - прыжок;

"вправо" - сальто вперед;

"влево" - сальто назад.

Повторное нажатие клавиш "вправо", "влево" балансирует прыгуна, то есть прекращает вращение и выравнивает его туловище перед приземлением.

BIATHLON - бег на лыжах со стрельбой по мишеням. После того, как Вы пробежите определенное расстояние, появится стрельбище. В Вашем распоряжении пять мишеней и пять выстрелов. В случае попадания, Вы увидите отверстие от пули в мишени. Цельтесь аккуратно, не торопитесь - ведь за каждый промах Вас оштрафуют на пять секунд.

Управление при ходьбе на лыжах:

"вправо" - шаг на ровной местности и при подъеме.

"вниз" - толчок палками при спуске.

Управление при стрельбе:

"вниз" - вставить патрон в патронник.

"вверх" - взвести курок.

"FIRE" - выстрел.

SPEED SKATING - скоростной бег на коньках. Вы бежите вместе с соперником.

Клавиши управления конькобежцем:

"вправо" - шаг правой ногой.

"влево" - шаг левой ногой.

SKI JUMP - Прыжки с трамплина. Вы должны выполнить в прыжке как можно больше акробатических элементов.

Клавиши управления прыгуном:

"FIRE" - начать спуск с трамплина.

"FIRE" (повторное нажатие) - отрыв от трамплина.

"вправо" - выполнение акробатической фигуры.

"влево" - выравнивание туловища перед приземлением.

FREE SKATING - фигурное катание. Время выступления ограничено. Фигуристка начинает двигаться сама. Вы должны выполнять различные упражнения фигурного катания.

Клавиши управления фигуристкой:

"влево"+"FIRE" - разворот фигуристки спиной.

"вниз" - вращение понизу.

"вверх" - вращение поверху.

"вправо"+"FIRE" - разворот фигуристки лицом вперед.

BOBSLEY - скоростной спуск в спортивных санях по ледяному коридору. Во время движения развивается огромная скорость, поэтому не злоупотребляйте нажатиями на клавиши - в любой момент Вы можете вылететь с трассы, а количество попыток ограничено! Клавиши управления санями:

"вправо" - качнуть сани против часовой стрелки.

"влево" - качнуть сани по часовой стрелке.

Продолжим рассматривать главное меню:

2. В этом случае Вам предлагается выбрать те виды спорта, в которых Вы хотите попробовать свои силы.

3. В этом случае Вам предложат поучаствовать в соревновании только в одном выбранном виде спорта.

4. Тренировка. Рекомендую начать именно с этого пункта, так как это самый верный путь узнать все тонкости управления спортсменом в выбранном виде спорта. Выбирайте и вперед, BEGIN на тренировку.

5. Вам предлагается выбрать количество игроков для совместных соревнований. Первоначально играет только один, а максимальное количество игроков - 4.

6. Предлагается выбрать удобное для вас управление:

1. Назначение клавиш.

2. Кемпстон джойстик.

3. Синклер джойстик-2

4. Курсор джойстик.

7. Вам предлагается просмотреть таблицу рекордов по тому или иному виду спорта.

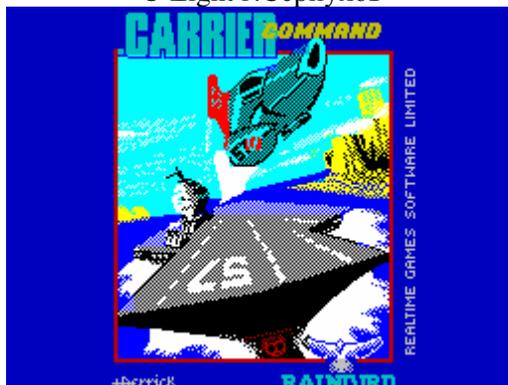
В заключение хочется сказать, что у Вас есть уникальный шанс побить мировой рекорд и стать чемпионом в любом из вышеописанных видов спорта. Ну, а если Вам удалось побить рекорды по всем видам спорта, представленным в игре - Вы просто гений! Даже неясно, почему Вас не включили в сборную?

...

Удачи Вам!!!

CARRIER COMAND

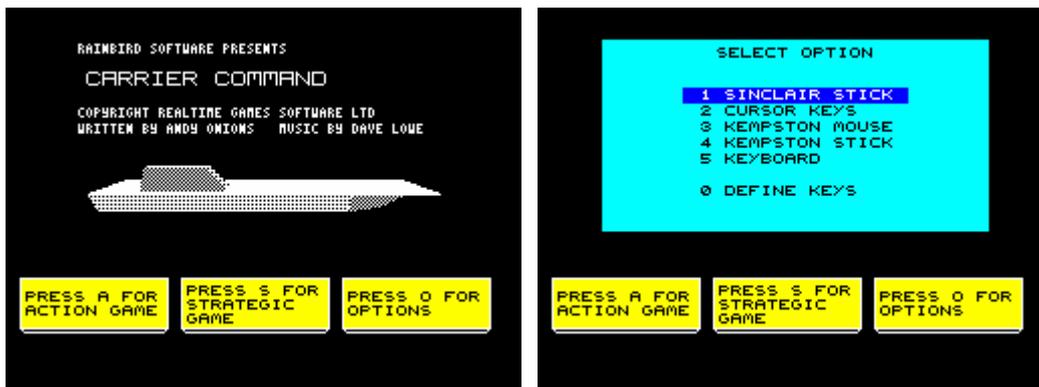
© Light г.Серпухов



CARRIER COMAND - замечательный симулятор авианосца. Он был произведен фирмой RAINBIRD SOFTWARE, и работает в режиме реального времени. Управление данной игрой происходит посредством пиктограмм, которых в ней очень большое количество, и разобраться в них сначала очень не просто. Однако, после ознакомления с ними, эта игра доставит Вам много часов, а то и дней, интересно проведенных с Вашим компьютером.

НАЧАЛЬНАЯ ЗАСТАВКА.

После загрузки игры, на экране появится вращающееся изображение Вашего корабля и название фирмы-изготовителя. Внизу экрана изображены три прямоугольника, в которых можно выбрать либо вид игры ("A" - ACTION, "S" - STRATEGIC), либо используемый вид управления (клавиша "O"). При нажатии на "O" перед Вами возникает меню:



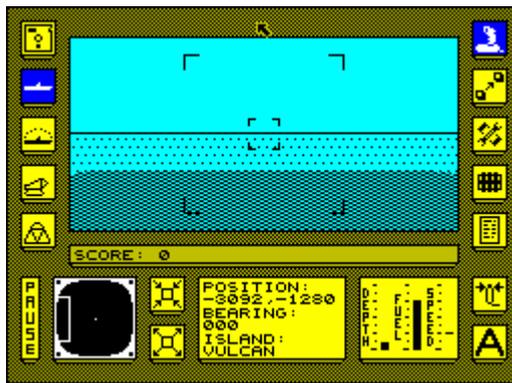
1. Синклер джойстик.
2. Курсор джойстик.
3. Кемпстон мышь.
4. Кемпстон джойстик.
5. Клавиатура.
0. Переопределение клавиш.

При выборе переопределения клавиш ("0"), перед Вами возникает новое меню, в котором Вам предлагают оставить их предыдущее значение ("Y"), а если Вы не согласны ("N"), то Вам предоставляется возможность их перераспределить. После этого Вы выходите в предыдущее меню.

Различие между видами игры STRATEGIC и ACTION состоит в том, что в STRATEGIC Вы начинаете игру с захвата нейтральных островов, тогда как при ACTION все острова уже распределены между Вами и противником.

ИГРА.

В начале игры перед Вами возникает вид из навигационной рубки корабля. По краям изображения расположены пиктограммы:



СЛЕВА: Сохранить/Загрузить игру.

Управление кораблем.

Управление защитной системой корабля.

Управление катером на воздушной подушке.

Управление самолетом.

Для выбора необходимой опции подведите к ней курсор и нажмите на "огонь".

СОХРАНИТЬ/ЗАГРУЗИТЬ ИГРУ.

Справа возникают четыре пиктограммы: сохранить игру, загрузить предыдущее состояние, выход (белый флаг), включить/отключить звуковые эффекты. Для подтверждения готовности выхода из игры необходимо нажать на клавишу "S".

УПРАВЛЕНИЕ КОРАБЛЕМ.

При выборе справа появятся следующие пиктограммы:

вид из капитанской рубки;

карта;

техническое обслуживание;

экономические характеристики базы;

сообщения;

ручное управление кораблем;

автоматическое управление кораблем.

ОПЦИЯ "вид из капитанской рубки".

Внизу возникают следующие пиктограммы:

ПАУЗА: игра останавливается и ждет нажатия клавиши "огонь".

ЭКРАН РАДАРА: показывает местоположение островов, заводов, катеров и т.д.

УВЕЛИЧЕНИЕ/УМЕНЬШЕНИЕ: увеличивает или уменьшает разрешающую способность экрана радара.

ДАННЫЕ О МЕСТОНАХОЖДЕНИИ:

местонахождение корабля (X,Y);

направление движения корабля (000 - направление на север; 180 - юг и т.д.);

название острова, находящегося в пределах досягаемости

локаторов, либо предыдущего острова.

ДАННЫЕ О ДВИЖЕНИИ:

глубина моря;

запас горючего;

скорость корабля.

При подходе к острову следует внимательно следить за показанием глубины, т.к. иначе Ваш корабль может налететь на мель и получить повреждения. Для перевода корабля на ручное управление необходимо навести курсор на опцию "ручное управление" и нажать на "огонь". После этого необходимо нажать на "контроль". Однако переход осуществим только при полной остановке корабля. Выход из этого режима - "контроль", опция "автоматическое управление"+"огонь". Увеличение или уменьшение скорости осуществляется подводом стрелки к опции "SPEED" + "огонь" (только для ручного управления). При автоматическом управлении скорость может регулироваться из опции "КАРТА".

ОПЦИЯ "Карта".

После ее выбора на экране изображается карта островов и местонахождение Вашего корабля. Передвигать карту можно с помощью пиктограммы в виде стрелок, а менять ее масштаб - пиктограммами с расходящимися или сходящимися стрелками. Пиктограмма со знаком вопроса покажет местонахождение Вашего корабля (при выбранной пиктограмме "STRATEGY" либо Вашей базы ("RESOURS")). Для нахождения ближайшего к Вашему курсору острова, нужно нажать клавишу "контроль".

Для прокладки маршрута корабля необходимо при выбранной пиктограмме "STRATEGY") подвести курсор к нужному острову и нажать "огонь". При этом на том месте появится крестик. После этого надо выбрать опцию "PROG" и маршрут будет запрограммирован. В случае отмены данного маршрута необходимо выбрать "CLEAR". Для ускорения времени пользуйтесь пиктограммой с изображением часов. Выбор новой базы аналогичен, но должна быть выбрана опция "RESOURS".

ОПЦИЯ "Техническое обслуживание".

Показывает состояние корабельного оборудования (на экране оно выделяется повышенной яркостью):

Корпус - может сильно повреждаться при столкновении с островом.

Радар.

Надстройка.

Лазер.

Ракетная установка.

Посадочная площадка.

Двигатели.

Со временем корабль восстанавливает поврежденные механизмы, но для быстрого ремонта применяется пиктограмма с изображением корабля и гаечного ключа. Скорость восстановления отдельных видов оборудования можно регулировать (нижняя левая пиктограмма).

ОПЦИЯ "Экономика базы".

Внизу расположены следующие пиктограммы:

Оборудование и вооружение вашего корабля.

Оборудование и вооружение базы.

График производства.

Максимальное производимое количество.

Режим погрузки на корабль.

Изначально показатели на графике производства стоят на минимуме. Для их переустановки надо установить курсор на нужном режиме нужного пункта и нажать "огонь".

Для указания количества производимых изделий необходимо также выбрать вид изделия (при этом он выделится), а затем, выбирая курсором знак увеличения или уменьшения, выставить необходимое значение.

Режим погрузки на корабль аналогичен предыдущему режиму, но изменение происходит при непосредственном указании на изделие.

ОПЦИЯ "Сообщения".

Выводит различную полезную информацию.

УПРАВЛЕНИЕ ЗАЩИТНОЙ СИСТЕМОЙ КОРАБЛЯ.

Возможны два варианта - непосредственное управление орудием и установка защитных роботов.

После включения опции "установка защитных роботов" (вторая опция справа), Вы увидите в нижней части экрана следующие пиктограммы: переключение роботов (робот 1, робот 2). С помощью курсора можно устанавливать их расположение относительно корабля. Впрочем, вторая снизу пиктограмма позволяет расположить их по заранее заданной схеме. И в том, и в другом случае необходимо затем указать на пиктограмму "PROG". Остальные две пиктограммы - это выпуск и отзыв роботов.

Управление орудием (правый верхний угол) осуществляется путем наведения прицела с помощью пиктограммы в виде курсора. Изображение цели можно увеличивать или уменьшать, при этом на экране будет дана кратность увеличения. Переключение вида оружия (ракеты или лазер) осуществляется путем выбора пиктограммы с нужным изображением. Лазер при стрельбе сильно разогревается, о чем свидетельствует пиктограмма в правом нижнем углу. При достижении критической температуры лазер автоматически отключается.

УПРАВЛЕНИЕ КАТЕРОМ.

Изначально на корабле находятся три катера на воздушной подушке, которыми Вы можете управлять. И еще три в виде запчастей. Катера применяют как для поставки на остров оборудования, так и для ведения военных действий, поэтому на него можно загружать как технику, так и оружие. Для этого надо выйти в меню "техническое обслуживание", заправить катер (курсор на "FUEL"+"огонь") и загрузить в него (окна в изображении катера) все необходимое. Выбор оборудования производится пиктограммами "+" и "-", "X" - отмена. Также, там показывается условное обозначение выбранного оборудования, а ниже его характеристика: код, вес, количество, наименование. После оснащения катера его выводят в док (изображение корабля с выходящей из него стрелкой). Для принятия корабля из дока служит аналогичная пиктограмма со стрелкой внутрь корабля. Из дока корабль выводится с помощью пиктограммы "работа с доком" (четвертая справа). При этом появятся дополнительные пиктограммы, которые позволяют либо выпустить корабль, либо принять его обратно.

Работа с картой для катера аналогична работе с картой для корабля.

РУЧНОЕ УПРАВЛЕНИЕ КАТЕРОМ.

Выбирается пиктограммой в правом верхнем углу. На экране появляется обзор из катера. Для перехода на ручное управление надо выбрать шестую справа пиктограмму и нажать "контроль". Скорость можно задавать либо подведя курсор к опции "SPEED", либо клавишами "faster"- увеличение и "slower" - уменьшение. Для сброса груза необходимо при выбранной опции данного груза, и при ручном управлении, нажать "огонь".

УПРАВЛЕНИЕ САМОЛЕТОМ.

Воздушная атака - очень удобный вид уничтожения противника. Управление самолетом аналогично управлению катером. Однако надо внимательно следить за показаниями высоты (ALT) и топлива (FUEL). Дозаправка самолета возможна как на корабле, так и на острове (даже на вражеском). Для приземления на остров необходимо наличие на нем аэродрома. При подлете к посадочной полосе, необходимо придерживаться ее середины, а также, сбросить скорость и высоту до минимума. Если все сделано правильно, то через некоторое время управление перейдет в автоматический режим, и дальнейшая посадка пройдет без Вашего участия. Для взлета необходимо подвести курсор к пиктограмме "автоматическое управление + "огонь", либо подать с корабля команду для возвращения: "работа с посадочной площадкой" (четвертая справа) + "возвращение на корабль". Аэродром на острове вмещает три самолета, но оставлять их там не рекомендуется, т.к. при переходе корабля к другому острову, они будут считаться разрушенными. При возвращении назад Вы их уже не увидите.

КРАТКАЯ ХАРАКТЕРИСТИКА ОРУЖИЯ.

ASSASIN - авиаракета с самонаведением. Довольно легкая, поэтому рекомендуется применять как дополнение к более мощному и тяжелому оружию.

ASSALING - аналогична предыдущему оружию, но включает в себя три ракеты, с соответствующим увеличением веса.

HARBINGER(2,3) - ракеты для катера. Удобны тем, что имеют самонаведение. Однако, в связи с тем, что в катере помещается максимум три ракеты (HARBINGER3), то серьезного ущерба неприятелю они не причиняют.

HAMMERHEAD - ракеты для корабельной артиллерии. Полезны для уничтожения противника из корабля.

QUAKER - авиабомба. Обладает наибольшим разрушительным воздействием.

QUASAR - маломощный химический лазер. Устанавливается на самолете и обладает неограниченным количеством выстрелов.

AVATAR - маломощный пульсирующий лазер. Устанавливается на катере и имеет ограниченный запас энергии.

DRONE - защитные роботы. Хорошо помогают в бою с кораблем противника.

THE GHOSTBUSTERS 2

(ACTIVISION, 89)

Эксперт: Александр Поршин, г.Арсеньев, Приморский край.

После загрузки основного блока заиграет музыка, и появится надпись

«PRESS FIRE TO START WITH KEYBOARD».

«PRESS FIRE TO START WITH JOYSTIC».

Это означает, что играть Вы можете как с помощью джойстика, так и на клавиатуре. Управление от клавиатуры:

J - вверх

N - вниз

Z - влево

X - вправо

K - огонь

Игра поддерживает KEMPSTON JOYSTICK и SINCLAIR 1.

Уровень 1.

Вам предстоит, управляя одним из охотников, спуститься на канате в вентиляционную шахту метро станции VAN HORNE, наиболее густо населенной приведениями, и собрать плазму на дне шахты. Но охотник уронил три составные части устройства, которым он должен собрать плазму. И поэтому, спускаясь, нужно собрать эти части, которые благополучно застряли в нишах шахты.

Управление охотником:

Влево - вправо - раскачивание на канате.

Огонь + влево - вправо - вращение вокруг своей оси.

Вверх - вниз - смотать/размотать канат.

Огонь + вверх или вниз - направление стрельбы.

На этом уровне охотнику постоянно мешают приведения, которые уменьшают его храбрость (COURAGE). Постоянно существует угроза обрыва каната, к которому пристраиваются точильщики. Руки и лица на стенах пытаются схватить его, уменьшая тем самым его храбрость.

Приборная панель:

WEAPON - вид оружия. Их всего три: лазер, лазерная перегородка, энергетический щит.

Переключение оружия осуществляется нажатием SPACE. В нишах можно подобрать заряд для любого оружия.

DEPTH - указатель длины каната в метрах.

CABLE - указатель состояния каната.

COURAGE - показатель храбрости охотника. Если он кончится, то охотник отпустит руки и упадет.

В черном окне будут появляться составные части устройства для сбора плазмы.

Уровень 2.

Охотники, используя собранную плазму, оживляют статую свободы и с ее помощью пытаются пробиться к музею. На часах без четверти двенадцать, а ровно в полночь из картины в музее выйдет молдавский черт Виго, и игра закончится плачевно. Вы управляете маленьким приведением, Лизуном, который летает перед статуей и расстреливает злых призраков. Энергия статуи изображена в виде баночки в нижней части экрана. При столкновении статуи с призраком, ее энергия уменьшается, но ее можно пополнить. Делается это следующим образом: при поражении злого духа от него на земле остаются сгустки плазмы. Если Вы нажмете SPACE, то три охотника, которые бегут за статуей, выйдут вперед и подберут плазму для статуи.

Уровень 3.

Вы в музее, но картина находится в подвале, а попасть туда можно только через дыру в потолке. Во время спуска у охотника работает только клавиша "вниз".

Подвал изображен в трех измерениях, он занимает два экрана. Когда все охотники благополучно доберутся в подвал, необходимо выполнить следующие действия:

1. Бегите во второй экран. Там, перед картиной, на маленьком жертвенном столике, лежит ребенок. Подбегите к нему.

2. Нажмите ENTER. Вы увидите лица охотников, а под ними – тип оружия. Вы сейчас управляете Реем. Наведите стрелку на его оружие, нажмите "огонь" и перемести те оружие кому-нибудь другому. Снова ENTER.

3. Возьмите ребенка, и бегите в первый экран, где и оставайтесь.

4. Нажав два раза на SPACE, переключитесь на Игона. Как Вы заметили, здесь бегают какой-то тип и стреляет в Вас. Пора его прикончить.

5. После его гибели из картины выскочит Виго. Не давайте ему пробиться к Рею с ребенком.

6. Убить Виго можно, стреляя из двух лазеров сразу.

7. Когда Вы уничтожите Виго, его дух вселится в Питера. Помочь ему можно только одним способом - засунуть Питера с помощью лазера в пустой холст.

Все эти действия нужно выполнять молниеносно.

HACKER

Эксперт: Олег Бобрышев, г. Георгиевск.

Итак, в этой программе Вы выступаете в роли взломщика (хакера) компьютерной сети. В самом начале Вам предлагают ввести код доступа. Откуда Вам его знать? После нескольких, очевидно, неудачных, попыток предлагают помощь и сообщают, что новый код - секретное расположение испытательной площадки. Час от часу не легче! Продолжаем бесполезное угадывание, после чего нас выключают из сети. Но... система безопасности неожиданно отключается, и Вас приглашают в концертн Магма, ЛТД.

Здесь Вам сразу же предлагают подземное дистанционно управляемое устройство (SRU), иначе говоря, робота. Модель номер AXD-0310479, пригодная для секретных операций.

Затем предлагают пройти тест. Чтобы его пройти, необходимо навести крест на тестируемое устройство и нажать "огонь". Тестируются следующие устройства:

Инфракрасный сенсор видеоизображения - на "голове" у робота.

Асинхронный уплотнитель данных - "клешня"

Гидравлический привод - выделен цветом.

Соединение Фламзона (FLAMZON JOINT) - центр заднего колеса.

Порт Тельмана (THELMAN PORT) - выступ "на спине".

Если Вы не нашли эти устройства с первого раза, Вам предложат пройти тест повторно, иначе Вы получите сообщение: "Диагностический тест пройден. Ошибок не найдено."

После этого Вы попадаете на главную карту. Здесь Вас попросят ввести имя, а затем установить текущее время. Зачем это делается, я так и не понял, хотя есть предположение, что на ночной стороне Вас реже замечают спутники-шпионы.

После того, как Вы закончите настройки, Вам сообщат, что робот находится в Южной Атлантике, и начнется основная игра. Вы управляете роботом, намечая его маршрут по подземным тоннелям. Рекомендуются с первых нескольких игр составить карту этих тоннелей, чтобы потом не путаться.

Кстати, составляя карту, я обнаружил, что в Австралию попасть нельзя. Как только Вы приблизитесь к ней, получите сообщение: "Вы вошли в охраняемую зону. Всем дистанционным устройствам было приказано покинуть испытательную площадку.

Согласно директиве Ваш модем будет отключен.

До свидания"

Это, естественно, конец игры, но он позволяет сделать вывод: испытательная площадка находится в Австралии!

Проверяем: На начальный вопрос отвечаем AUSTRALIA, и, минуя тесты, сразу же оказываемся на стадии установки времени!

Вообще, как я заметил, время в этой игре считается ходами. То есть, все действия (конкретно - выдача сообщений) происходит через конкретное количество ходов, если брать за один ход перемещение на один пункт сетки карты. Сообщений, предусмотренных на определенном ходу, всего шесть. Привожу их в своем переводе. Каждое из сообщений озаглавлено MAGMA LTD. PROJECT UPDATE.

Сообщение N1. Выдается на старте:

Последняя целевая скважина достигла заданной глубины. Успешное управление температурой поверхности плазмы обеспечивает неограниченный запас энергии и мировое господство.

Внимание! Сейсмические отражения указывают на возможность выхода тепла из-под контроля, что приведет к разрушению континента. Всем устройствам остается только покинуть испытательную площадку!!

Сообщение N2. Выдается на десятом ходу.

Штаб-квартира разведки докладывает, что шпионы украли части измельченного документа, содержащего частную информацию концерна "Проект Магма".

Ваша миссия: Используя робота для торговли с этими шпионами, вернуть все части документа до того, как эти доказательства достигнут федеральных агентов. Вы начинаете со стандартной суммой денег для операций такого рода. Вы должны купить доказательства, или купить товар, который можно обменять на доказательства.

Предупреждение!

Хотя разведка точно не определила, чего требует каждый шпион в обмен на свой кусок доказательства, но известно, что некоторые шпионы могут принять предметы, которые требуются другим шпионам.

Что я могу к этому добавить? Это точная и исчерпывающая инструкция к игре. С одним "но". Поскольку информация предназначалась сотруднику корпорации, а не хаккеру, Ваша задача, очевидно, прямо противоположна - доставить улики федеральным агентам.

Сообщение N3. Выдается на 28-м ходу.

Внимание: Колебания энергии в реакторе повредили сопротивление смещения реактора смещения.

Ожидается работа с перебоями дальнедействующего сканера.

На практике это означает, что Вы по-прежнему можете видеть, на какую клетку Вы навели сканер, но скорее всего, не узнаете, какие выходы ведут из этой локации, пока сами туда не прибудете.

Сообщение N4. Выдается на 55-м ходу.

Тревога! Вторжение!

Обнаружена брешь в системе безопасности. Источники показывают, что в подземную систему тоннелей проник враг. Доказательства, угрожающие существованию "Проекта Магма" не должны достигнуть Вашингтона, округ Колумбия.

Секретные спутники активированы.

После этого над картой начинают летать далеко не безобидные спутники. Но о них - позднее.

Сообщение N5. Выдается на 96-м ходу.

Предупреждение!

Как следует из доклада о повреждениях, в блоке дистанционного слежения произошел фазовый сдвиг.

Система отказала. Продолжайте осторожно.

После этого сообщения сканер отказывает вообще. Свое место нахождения Вы можете определять только по памяти. Вот где сильно пригодится составленная карта!

Сообщение N6. Заключение. Выдается на 181-ом ходу.

Вы превысили необходимый лимит времени.

Из-за недавней тревоги, вызванной вторжением, Вы считаетесь угрозой безопасности. Ваш модем отключен.

До свидания.

Игра после этого, естественно, заканчивается. Таким образом, необходимо выполнить миссию менее чем за 181 ход.

Итак, задача ясна. Необходимо посетить десять городов по всему свету и затем предоставить полученные доказательства в одиннадцатый город - Вашингтон.

Во время игры Вам будут мешать спутники безопасности. Впрочем, для умелого хаккера они - не помеха. Вот их запросы:

Первый уровень безопасности: Назовите имя кампании. Ответ очевиден - MAGMA, LTD. Обратите внимание, что после запятой обязателен пробел, а после сокращения LTD - точка.

Второй уровень безопасности: Назовите модель робота: Как известно, это AXD-0310479.

Третий уровень безопасности: Назовите тип привода - гидравлический (HYDRAULIC).

Четвертый (максимальный) уровень безопасности: Назовите местонахождение испытательной площадки. Как я уже выяснил, это - Австралия. После четвертой проверки Вас уже не тронут - можете спокойно торговать.

Торговля происходит по следующему принципу: каждому шпиону Вы должны принести нужный ему предмет. В обмен шпион отдает Вам один из кусков документа, а затем предложит два предмета на продажу. Вот список локаций с теми предметами, которые там предлагают:

1. Турция (видимо, Стамбул)

- STAR OF INDIA (Алмаз "Звезда Индии"), 5000\$.
JEWELLED LAMP (Драгоценная лампа), 1000\$.
2. Греция
ANCIENT ARTIFACT (Предмет старины), 1000\$.
GRECIAN URN (Греческая урна), 1000\$.
3. Египет
EMERALD SCARAB (Изумрудный скарабей), 1500\$.
GOLD STATUE OF TUT (Золотая статуэтка бога Тут), 1000\$.
4. Франция
DEEDS TO A SWISS CHALET (Права на шале в Швеции), 9500\$.
CHRONOGRAF (Часы), 200\$.
5. Великобритания (видимо, Лондон).
AUTOGRAPHED BEATLES ALBUM (альбом "Битлз" с автографом), 900\$.
CROWN JEWELS (Драгоценности короны), 9500\$.
6. США (видимо, Нью-Йорк).
UNCUT 3KT DIAMOND (неграненый алмаз весом 3 карата), 2000\$.
STOCK AND BONDS (акции и облигации), 2000\$.
7. США (видимо, Лос-Анжелес).
GOLD NUGGETS (Золотые самородки), 1500\$.
49-er SEASON TICKETS (49-ти сезонные билеты), 200\$.
8. Бразилия
SPANISH DOUBLONS (Испанские дублоны), 500\$.
TREASURE MAP (Карта сокровищ), 700\$.
9. Япония
CULTURED PEARLS (Выращенный искусственно жемчуг), 3000\$.
35-mm CAMERA (35 миллиметровая камера), 300\$.
10. Китай
MING VASE (Ваза династии Минг), 2000\$.
JADE CARVING (Резьба по жадеиту), 900\$.

Как видите, в наличии могут быть 20 предметов - есть где запутаться! Да еще ограничения по наличности, ведь у Вас только 35000\$. Кстати, каждому шпиону, кроме предмета, можно предложить еще и деньги. Но большая часть предметов здесь никому из шпионов не нужна, и сбыть их не удастся. Хорошая головоломка, чтобы над ней подумать. Решений у нее несколько, то есть, несколько возможных маршрутов движения, но большинство из них не укладывается в 180 ходов.

Мною был разработан кратчайший маршрут, однако приводить его на страницах журнала не имеет смысла, иначе пропадет всякий интерес. Подскажу только, что первой Вы должны посетить Францию – там шпион берет наличные. Далее следует цепочка городов, и серия обменов, причем вовсе не обязательно покупать дорогой предмет - здесь цена не является весомым фактором при выборе.

Когда вы доставите документы в Вашингтон, Вас поблагодарят и покажут передовицу газеты "Вашингтон пост". На этом игра заканчивается.

LAST BATTLE

Фирма: HACKERS SQUAD, Россия
Эксперты: Хромов С.В. (г. Ярославль)
Селиверстов Е.В. (г. Старый Оскол)

Предлагаем Вашему вниманию комплексное описание новой российской стратегической игры LAST BATTLE, которое мы составили на основе двух писем наших корреспондентов, дополняющих друг друга. Игра LAST BATTLE наверняка понравится любителям LASER SQUAD и REBEL STAR. Итак....

Ящеры Сириуса, разбитые в 729 году галактической эры, снова мечтают о создании империи. Они проникли в недавно открытую систему, где на шести планетах был сосредоточен весь военно-промышленный комплекс давно исчезнувшей цивилизации. Базы, находящиеся на планетах, создают роботов-монстров, способных разбить объединенные силы Земли.

Однако, об этом стало известно, и руководство Земной Федерации отправляет Вас с заданием ликвидировать опасность.

На этом и начинается игра. Под Ваше руководство дается одна база, захваченная землянами. Вы можете производить шесть типов роботов на своей базе. Кроме того, у Вас уже есть два слабых робота, способных захватывать базы. Для полной победы над врагом Вам надо захватить все базы противника, производящих роботов захвата и избавиться от всех роботов захвата.

При загрузке игры Вы увидите заставку с меню, в котором можно выбрать:

- 0 - игра;
- 1 - инструкция;
- 2 - редактор.

Не будем останавливаться на двух последних пунктах, расскажем об управлении игрой.

После выбора этого пункта меню, Вам предложат занять командный пункт, и появится меню:

- EXIT - выход в главное меню;
- DONE - перейти к игре;
- LOAD, SAVE, VERIFY – дисковые операции.

При выборе дисковых операций появится возможность выбрать одну из шести отгрузок.

Если Вы выбрали DONE – увидите краткую оперативную сводку, в которой показывается: время, прошедшее с начала операции, количество баз, способных производить роботов захвата и общее количество баз; количество роботов захвата и общее количество роботов.

Тут надо сказать, что в игре шестнадцать типов роботов, из которых пять типов способны захватывать базы. Роботы разделяются по типу передвижения на сухопутные и морские. Сухопутные могут передвигаться по суше и береговой линии, а морские - по морю и по берегу. Они в свою очередь, делятся еще на два вида: сухопутные - на боевых и ремонтных, а морские - на боевых и транспортных. Кроме того, роботы различаются по количеству ходов и защитной энергии.

Вернемся к управлению игрой. Нажмите "огонь" и перед Вами возникнет участок карты с имеющейся у Вас базой. Все пространство вокруг пока закрыто черным цветом, который Вам предстоит очистить в процессе захвата планеты. Справа от карты Вы видите справочную таблицу, из которой можно узнать:

MAP - какой элемент карты находится под курсором.

OBJECT - объект, находящийся на элементе карты.

APS - количество шагов робота.

SHIELD - состояние защиты.

NUMBER - номер робота.

STATUS - статус объекта: нейтральный/чужой, для своих баз - пустая/количество роботов, для своих роботов - назначение робота.

Наведите курсор на базу, и нажмите "огонь". На месте таблицы появится меню управления:

1 ORDER - программирование базы.

2 LEAVE - выбор робота. Если на базе несколько роботов, то этой опцией Вы можете выбрать одного из них для управления.

3 SELECT - режим управления роботом.

4 NEXT TOWN - переход на следующую базу (можно совершить переход и по другому, нажав клавишу "B").

5 NEXT UNIT - переход к следующему роботу (клавиша "N").

6 OPTIONS - изменение параметров игры.

7 REPORTS - рапорт о сражениях, прошедших за ход.

8 SCANNER - карта планеты (клавиша "S").

9 CANCEL - возврат в режим перемещения по карте.

10 END TURN - передача хода противнику.

Большинство пунктов в этом меню имеют свои вложенные опции.

1 Режим управления базой. ORDER:

а. CANCEL - выход из режима программирования.

б. CHANGE - вызов вариантов тех роботов, которых может производить база вместе с краткой характеристикой каждого из них.

в. FORBIDEN - прекратить выпуск роботов.

г. GROWTH (REDUCE) - порядок защиты базы роботами, находящимися на ней - слабые (сильные).

д. INFORM - информация о выбранном роботе. Если при входе в это меню база уже производит какого-нибудь робота, Вы увидите его название, краткую характеристику и время до выпуска из производства.

3 Режим управления роботом.

SELECT:

а. END MOVE - конец управления.

б-г. OPTIONS, REPORT, SCANNER - аналогично описанным выше опциям.

д. UNLOAD - выгрузит робота с транспорта (присутствует только у транспортов)

е. UNPACK - разворачивание мастерской в рабочее положение. Теперь получившему повреждение роботу достаточно встать на нее, и начнется ремонт.

ж. PACK - сворачивание мастерской.

з. EXPLODE - взорвать робота вместе с базой.

и. UTILISE - разрушить робота. Во время игры Вы можете создать только 170 роботов и 30 транспортов, поэтому имеет смысл уничтожать ненужных и слабых роботов и заменять их более сильными.

к. STRIP - привести робота на базу, ранее не производившую этот тип, Вы можете разобрать его на детали и начать производство. База может производить до восьми типов роботов.

л. COURSE - установка курса роботу.

м. CANCEL - возврат в режим перемещений.

6 Режим изменения игры. OPTIONS:

а. CONTROL - тип джойстика (SINCLAIR/KEMPSTON)

б. MOTION - тип управления роботом. Делится на два вида:
ROTATION - клавиши "влево"/"вправо" - поворот робота, "вперед"/"назад" - перемещение.
DIRECT - перемещение, указывая направление клавишами или джойстиком.

в. COURSE - тип движения по карте:
CALL AT - с заходом на свои базы.
PASS BY - запрет на заходы.

г. SCROLL - тип перемещения по карте при выборе опции NEXT TOWN / NEXT UNIT:
SMOOTH - плавно
IMMEDIATE - мгновенно.

д. BATTLE - вид демонстрации битвы:
FULL - полный бой
FAST - только итог.

е. SOUND - включить / выключить звук.

7 Рапорт о боях. REPORTS:
Выбрав эту опцию, Вы попадете в таблицу, где отмечены:
WON - битвы, выигранные Вами.
LOSS - проигранные сражения.
LOSSES IN TOWN - захваченные у Вас базы.
LOSSES IN ACTION - роботы, разбитые в бою.
LOSSES IN DISMANTLE - роботы, разбитые во время ремонта или транспортировки.

8 Режим карты. SCANNER:
Здесь Вы можете выбрать, кого Вы хотите увидеть на карте:
ALL - все войска и базы.
OWN - только свои.
ENEMY - только чужие.

10 Передача хода. END TURN:
Эта опция дает возможность ускорить ход игры. Если Вам надо быстро построить роботов или дойти к нужному месту, то появится меню:
END TURN - конец хода.
CANCEL - возврат.
ADSENT - выбор быстрого режима.
Выбирайте ADSENT. Вам будет предложено выбрать один из двух режимов:
FIRST DEMAND - игра не останавливается до первого события, которое требует Вашего вмешательства (выпуск робота, достижение поставленной цели, невозможность прохода, нападение противника).
DAYS AMOUNT - на несколько дней (0-9).
Режим установки курса роботу. COURSE.
EXIT - выход.
GIVE - задать курс; для этого подвести курсор к нужному месту и нажать "огонь".
CLEAR - очистить память робота от программы движения, робот переходит на ручное управление.
VIEW - посмотреть курс.
При движении робота по курсу могут появляться следующие сообщения:
GOAL IS GAINED - цель достигнута.
SPOTTED THE ENEMY - замечен враг, можно либо на него напасть, либо ретироваться.
WAY IS BLOCKED - путь закрыт. Надо либо вручную провести робота, либо очистить координаты.

Вот и все об управлении. Эта игра имеет шесть уровней сложности и может понравиться как начинающим стратегам, так и гениям военного дела. Как пишут сами авторы, на уровне SUPER BRAIN игру не смогли пройти даже они. Между тем, чтобы выиграть на последнем уровне сложности вовсе не обязательно быть "или параноиком или гением" (цитата из авторского описания), достаточно четко придерживаться некоторых правил управления своими войсками.

Для начала нужно как можно быстрее исследовать собственный материк, отыскать нейтральные базы, захватить их и наладить выпуск роботов. Постепенно на тех базах, где выпускают простые модели нужно развернуть производство более мощных и маневренных ASSASSIN'ов.

Теперь можно взяться и за другие материки и острова. Хорошо бы быстро обнаружить месторасположение противника, чтобы в дальнейшем следить за его действиями. Также быстро надо обнаружить и захватить базы, которые производят ARMoured PREDATOR'ов – самых мощных наземных роботов.

Для сравнения: PREDATOR имеет 24 единицы защиты и делает 2 шага за сутки, а простейший имеет три единицы защиты и делает 2 шага за сутки. База (любая) имеет 3 ед. защиты, ASSASSIN - 6 ед. защиты и три шага/сут. BUTCHER - 10 ед. защиты и 1 шаг/с. Против PREDATOR'а эффективно сражаться может только BUTCHER, зато этого самого BUTCHERA "заваливают" двое ASSASSIN'ов.

Если враг обнаружен, рекомендуем на том берегу Вашего материка, который расположен ближе всего к частям противника, сосредоточить группу роботов, состоящую из простых ASSASSIN'ов ("на убой") и парочки роботов покрупнее. Это будет команда быстрого реагирования на случай высадки вражеского десанта. Десант обычно высаживается численностью в пять или восемь роботов (так как существует только два вида транспортных кораблей).

Некоторые пограничные базы могут по несколько раз переходить из рук в руки. Иногда это и к лучшему - ведь захватив вновь свою базу, Вы можете обнаружить, что инопланетяне наладили на ней производство роботов с лучшими показателями. Но если база по каким-либо соображениям не должна оказаться в руках врага (например, на ней скоро должен сойти с конвейера новый PREDATOR, а отстоять ее нет никакой возможности), то лучше эту базу взорвать.

Обнаружив противника, постарайтесь разведать где находятся его морские базы - их обычно немного и они располагаются на берегу. К этому времени у Вас должна быть хотя бы пара MEGAZAYR'ов или кораблей поменьше. Этими кораблями нужно занять позицию у морских баз так, чтобы видеть любой вражеский корабль, вышедший из ее ворот и уничтожить его. Теперь надо устроить охоту на уже созданные корабли противника (в особенности на транспортники). Одновременно с этим, загружаем в транспорт на восемь мест ASSASSIN'ов и направляем их к морским базам противника. Совсем необязательно, захватив морскую базу, удержать ее. Достаточно продержаться там сутки, после чего загружать всех уцелевших роботов обратно на судно, оставив в базе самого избитого, и вместе с ним взорвать ее к чертовой матери. С остатками отряда двигаемся к следующей базе, где делаем то же самое. После того, как все базы, способные производить транспортники будут взорваны, а сами транспорты - утоплены, можете считать, что победа у Вас в руках. Теперь можно не спеша накапливать десантные отряды вдоль побережий, пока не иссякнут ресурсы планеты, а это произойдет довольно скоро. К тому времени Ваша армия по численности и мощи в три-пять раз превзойдет армию врага! Только представьте себе самый быстрый и маневренный отряд из восьми ASSASSIN'ов. 3-4 таких отряда - и континент Ваш. Или восемь PREDATOR'ов - против такой толпы могут устоять лишь два смешанных отряда по восемь роботов, состоящие из BUTCHER'ов и MARAUDER'ов. Можно еще организовать отряды из семи BUTCHER'ов и одного PREDATOR'а в качестве командира. Также, не помешает ремонтная бригада из пяти ANIMTOR'ов и трех ASSASSIN'ов для охраны. Если хорошенько постараться, то за 50-60 дней можно уничтожить кого угодно.

Перед высадкой на планету можно посмотреть ее ландшафт, запомнить, где находятся острова и материки (базы противника генерируются случайно). Думаем, это не противоречит идее игры - Вы ведь можете увидеть это из космоса.

CAPTAN FIZZ

Олег Смолянкин из г. Киселевск Кемеровской обл. прислал нам русифицированную версию этой игры и полное описание к ней. Итак,



Сюжет

События происходят на звездолете "Икар", который потерял управление из-за поломки бортовых компьютеров. В результате этой поломки произошел сбой в программе управления кораблем, и теперь "Икар" летит к Солнцу, а значит - к смерти. Чтобы этого не случилось, нужно телепортироваться на борт корабля, уничтожить компьютеры и взять управление в свои руки.

Настройка программы.

В процессе загрузки Вам предстоит выбрать язык, на котором программа будет с вами общаться: английский, французский или немецкий (в оригинальной версии), или: английский, французский и русский в русифицированной. Если у Вас компьютер с памятью 128К, то вся музыка и звуки будут слышны через АУ. Управление игрой:

1-й игрок: SINCLAIR II или KEMPSTON

2-й игрок: SINCLAIR I или Z, X, N, J, SPACE.

Дополнительные клавиши:

M - вкл./выкл. музыки;

Q - выход из игры;

P - вкл./выкл. паузы;

A - супер-бомба первого игрока;

B - супер-бомба второго игрока;

Экран игры

Главный экран разделен на две части - верхняя для первого игрока, нижняя - для второго. Слева у каждого игрока есть информационное табло. Сюда выводится:

ARMOUR - Ваша защита.

DAMAGE - вред, наносимый Вашим оружием врагу.

CHARGE - заряд бластера.

CREDIT - количество "денег".

Еще сюда выводятся Ваши очки и температура Вашего оружия. На экране у обоих игроков выводятся:

HEALTH - Ваше здоровье

CARDS - карточки для доступа в другие отсеки корабля.

Кроме того, в правой части экрана показаны:

EXIT - выход.

SWITCHES - индикаторы отключения энергетического поля.

MINES - индикаторы мин.



Игра.

Сразу после телепортации ищите компьютеры (белый квадрат с буквой L) и уничтожайте их. Это не так просто, ведь все компьютеры имеют защитное поле и не с первого выстрела уничтожаются.

В игре встречаются некоторые полезные (и необходимые) предметы:

CARDS (пропуски) - цветные прямоугольники. Во многих уровнях есть двери, которые открываются только при наличии пропуска. Он должен быть такого же цвета, как и дверь.

Кроме того, есть и другие предметы - так, объекты зеленого цвета увеличивают очки, красные пистолеты увеличивают заряд бластера, а квадраты с буквой "А" повышают разрушительную силу Вашего бластера. Кроме того, Вы можете найти зеленые точки ("деньги"). С их помощью можно зарядить энергию у специальных энергетических емкостей.

Во многих уровнях доступ к предметам закрыт энергетическим полем. Его можно отключить (на время). Ищите на уровне четыре белых кружка. Прикасаясь к ним в определенной последовательности, Вы отключите энергетическое поле - загорятся четыре индикатора. Теперь не медлите!

Также, на всех этажах корабля обитают свихнувшиеся роботы. Мирно разговаривать с ними я бы не советовал. Еще Вам могут встретиться установки, которые стреляют ракетами. Установку выстрелом уничтожить нельзя, а ракету можно лишь развернуть в другую сторону. Тут Вам пригодится супер-бомба, которая уничтожит на экране всех роботов и установку.

Итак, если Вы уничтожили все компьютеры (загорается индикатор "выход"), то ищите овальный колодец и заходите в него. Уровень пройден. Но впереди еще тяжелые битвы!

Удачи!

КОМПЬЮТЕРНАЯ НОВЕЛЛА

© А. Березин. г. Колпашево, Томской обл.

ИСТОРИЯ В МОМЕНТ ТВОРЕНИЯ.

(компьютерная новелла по игре MYTH фирмы SYSTEM 3, 1989)

1. (Нью-Йорк, 2012 г.н.э.)

Жаркий июльский день был в разгаре. Бродвей, как всегда в часы пик, представлял собою восхитительное зрелище: непрерывный поток автомобилей всех моделей и цветов, пестрый людской муравейник на широких тротуарах... Ничто не предвещало беды.

Эрик неторопясь шел, засунув руки в карманы джинсовой куртки и насвистывал незамысловатый мотив. До конца обеденного перерыва оставалось еще полчаса, и он мог позволить себе никуда не спешить.

Вдруг Эрик заметил какое-то смятение в толпе впереди себя. Несколько человек, идущих ему навстречу, развернулись и побежали прочь, другие застыли на месте, с ужасом глядя на что-то за спиной Эрика. Эрик поспешно оглянулся, и в этот миг острые когти впились в его плечо. Сильный толчок швырнул его на землю лицом вниз. Эрик перекатился на спину и увидел нападавшего. От неожиданности он вскрикнул: громадные кожистые крылья поддерживали в воздухе скрюченное тело, к которому были прижаты кривые когтистые лапы. Длинный, похожий на крысиный, хвост свисал на целый метр, а тонкую бордавчатую шею венчала... женская голова! Ее правильные и даже красивые черты были перекошены дикой яростью, багровый огонь полыхал в глазах, кудрявые волосы были растрепаны и спутаны. С губ монстра падала на асфальт красная кровавая пена.

Чудовище неподвижно висело в воздухе в трех метрах от Эрика; вдруг со злобным и страшным криком оно сорвалось с места и вновь бросилось на него. К счастью, первый испуг Эрика уже прошел, и он приготовился к обороне. Все еще лежа на спине, он с силой ударил кошмарное создание ногой в грудь, прямо между нацеленных на него лап. Удар отшвырнул монстра назад, и он едва удержался в воздухе. С трудом, выровняв полет, тварь развернулась и полетела прочь. Эрик поднялся и, преодолевая боль в плече, побежал в другую сторону...

2. (Нью-Йорк, 2012 г.н.э. неделю спустя)

Профессор Гаспар Адамсон поднялся со своего места и прокашлялся.

- Итак, господа, я думаю, ситуация ясна всем. Вы просите меня высказать свое мнение как руководителя экспертной комиссии, и я с удовольствием делаю это. Но сперва я напомню вам основные события минувшей недели.

Общество, состоявшее почти из одних только ученых мужей, приготовилось слушать и придвинуло к себе блокноты. Вездесущая пресса и телевизионщики включили свои диктофоны и нацелили на Адамсона зрочки камер. Конференц-зал затих и весь обратился в слух.

- В прошлую пятницу Нью-Йорк подвергся нападению летающих чудовищ; их описание совпадает с внешним видом мифических гарпий Древней Греции. В субботу в пригороде Манчестера в Великобритании фермеры обнаружили на своих полях несколько надгробных камней там, где их раньше не было. На камнях надписи на латинском языке, относящиеся к неизвестной эпохе, существовавшей, вероятно, еще до становления Рима... В Италии ходят слухи о каких-то живых мертвецах близ Неаполя...

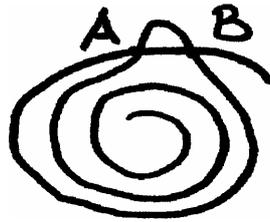
- Извините, профессор, - прервал Адамсона седой старик, ректор крупного французского университета, - Вас назначили выяснить причины появления чудовищ на Бродвее, а не собирать слухи со всего мира!

Адамсон махнул рукой куда-то в сторону.

- Терпение, терпение! Я все объясню. Итак... Во вторник пресса Греции опубликовала статью о человеке в наряде гладиатора, изъясняющимся на древнегреческом... Человек помещен в сумасшедший дом в Афинах... - Профессор сделал паузу и обвел взглядом конференц-зал. Комиссии под моим руководством удалось выяснить взаимосвязь между этими событиями. Я попробую объяснить все как можно понятнее. Адамсон подошел к доске и взял из жестяной коробки мел. Уверенными движениями он начертил на доске спираль.

- Время, как известно, представляет собой спираль. Движение во времени происходит от центра спирали к краям. Линия спирали нигде и никогда не пересекает саму себя - таков естественный ход времени. А теперь представьте себе, что где-то в далеком прошлом произошло нечто, что заставило спираль времени искривиться, и один ее виток пересек другой. Что мы тогда увидим?

Профессор быстрыми движениями накидал рядом с первой другую спираль - деформированную.



- Как видите, мы получили пересечение совершенно разных эпох. Такое пересечение может повлечь за собой частичное взаимопроникновение этих времен. Причем точек пересечения (взаимопроникновения) будет обязательно две. Я обозначил их буквами А и В. Точку А мы прошли неделю тому назад. Именно в момент соприкосновения разных эпох, нашей и древней неведомой эпохи, в наш мир проникли гарпии, гладиатор, надгробные камни... Теперь следует ожидать прохождения точки В, а с ней и повторного взаимопроникновения времен - появления людей и существ из прошлого, а может быть и из будущего. Если мы будем знать продолжительность интервала между точками А и В, то сможем точно установить, в каком времени произошло искривление спирали. Возможно, удастся даже выпрямить спираль... Но пока рано об этом говорить. У меня все. Спасибо за внимание.

Профессор опять откашлялся, положил мел в коробку и сел на свое место.

...Именно в это время в Норвегии величаво подходил к причалу странный корабль, украшенной на носу драконьей головой. На палубе его толпились люди в рогатых шлемах...

3. (Нью-Йорк 2012 г.н.э. вечер следующего дня)

- Ну вот, что я говорил! - потрясая газетой, кричал Гаспар Адамсон. - Вот она, точка В! Викинги в Норвегии! Самые настоящие! Приплывшие на драккаре... Что я говорил! Немедленно за расчеты!

4. (Нью-Йорк 2014 г.н.э.)

Профессор Адамсон разложил свои бумаги на трибуне, передвинул стакан с водой и по привычке откашлялся, как делал всегда перед публичными выступлениями. На этот раз аудитория его была куда больше, чем два года назад. Самый большой зал Института Времени едва вместил всех желающих послушать выступление профессора.

- Дамы и господа! Вам всем известно, что два года назад произошли неординарные события, повлекшие за собой частичное слияние минимум пяти эпох. Теперь мы - я и мои коллеги - можем с уверенностью сказать, какие именно точки в истории земли были затронуты в результате временной аномалии. Первое отклонение временной спирали от нормы произошло около 7 тысяч лет назад. Географически эта аномалия привязана к району нынешнего Неаполя. Виток времени по неизвестным причинам деформировался и задел соседний в точке примерно 400 года до нашей эры. Географическая точка соприкосновения - Греция. Далее виток два раза пересек наше время с интервалом в неделю, причем география соприкосновений была самая разнообразная - Нью-Йорк, Скандинавия, Англия... Возвращение витка к норме потревожило еще две эпохи: 500 год нашей эры в Древней Скандинавии и 3000 год до нашей эры в Древнем Египте. После прохождения этой последней точки временная спираль вошла в норму. Остается совершенно непонятным, что произойдет, когда время сделает еще один полный виток. Вполне возможен очередной контакт времен. Предсказать его последствия практически невозможно. Я считаю совершенно необходимым вернуть спираль в нормальное состояние. Уверен, что это можно сделать, устранив причину аномалии. Подготовка к экспедиции в прошлое идет полным ходом!

5. (Нью-Йорк. 2014 г.н.э. месяц спустя)

- Прошу Вас, постоянно помните о своей цели, не отвлекайтесь на мелочи, не пытайтесь вмешаться в ход истории. Никакого геройства! Слышите? От Вас зависит судьба всего мира и если Вы погибнете, завершить миссию за Вас будет некому...

- Я все понял, профессор. Будьте уверены, я не подведу Вас.

- Эрик... Я не знаю, сможете ли Вы вернуться обратно в наше время... Но я от всей души желаю Вам удачи, - Адамсон опустил глаза. Его душила злость на самого себя. Кто, как не он, должен был все досконально выяснить перед тем, как отправлять этого парня в прошлое, может быть на смерть?

"Да нет же, - отмахнулся от мыслей Адамсон. - Ты сделал все, что мог. Сделать больше твоя наука просто не в состоянии..."

Эрик, казалось, понял, о чем думал профессор. Он положил руку ему на плечо.

- Не расстраивайтесь! Я еще вернусь. Ну а если нет, это не Ваша вина. Я ведь доброволец... Я знаю об опасности этого путешествия и готов рискнуть.

Эрик развернулся и отправился в лаборантскую, где заканчивались последние приготовления.

Через полчаса легкий поворот рычага на пульте управления отбросил его на семь тысяч лет назад...

6. (Италия. 5048 год до н.э.)

Мелко дрожащая стрелка детектора темпоральной энергии указывала на восток. Эрик повернулся спиной к заходящему солнцу и, оглядываясь кругом, осторожно двинулся по улицам мертвого города. По сторонам улицы теснились белые руины, обломки колонн и гипсовых статуй валялись в пыли... Город давным-давно был покинут жителями, и стоял на высоком холме, словно сломанный зуб.

Дорога вывела Эрика к заброшенному, заросшему кустами и жесткой травой, кладбищу. Эрик поежился. Солнце садилось, и ему вовсе не хотелось провести ночь в компании мертвецов. Однако стрелка детектора указывала в эту сторону, а маленький красный огонек разгорелся сильнее - значит, он приближался к цели. Эрик вздохнул и перелез через низкую ограду из каменных блоков.

Через двадцать минут ноги его устали продираться через цепкую путаницу трав, и он присел на край широкого надгробия, чтобы передохнуть. Сняв с пояса металлическую фляжку, Эрик жадно глотнул воды и огляделся вокруг. Солнце уже почти зашло. Меньше половины его диска виднелось из-за города на холме. Поглядев на восток, Эрик решил, что если поторопится, то успеет достичь конца кладбища до темноты - ограда уже виднелась невдалеке. Обрадованный этим открытием, Эрик бодро вскочил на ноги... и земля разверзлась под ним. Судорожным движением рук он попытался ухватиться за край плиты, но пальцы его соскользнули, и Эрик покатился вниз, в темноту. Ударившись обо что-то твердое затылком, он потерял сознание...

...Сначала вернулась боль и ощущение разбитости. Затем - зрение. Над ним нависал низкий каменный потолок, с которого словно клыки неведомого чудовища, свисали сталактиты. Эрик недоуменно разглядывал их, пока к нему не вернулась память. Он вспомнил свое недавнее падение в узкую черную шахту, город, кладбище - и вскочил на ноги. Тело тут же отозвалось острой болью. Целую минуту Эрик боялся пошевелиться, затем осторожно ощупал себя. Кажется, ничего не было сломано, а многочисленные синяки и ссадины хоть и причиняли неудобство, но не очень волновали Эрика.

В подземелье проникал бледно-серый свет, но определить, откуда он исходит Эрик не смог. Под ногами его была большая каменная плита. Эрик прислонился к стене и, размышляя, что делать дальше, разгребал ногой песок и камешки. Вдруг нога его попала в углубление, показавшееся ему неестественно круглым. Эрик опустился на колени и смел рукой песок с плиты. Он оказался прав: в плите были вырезаны пять крупных полусферических углублений, расположенных крестом - четыре по краям и пятое в центре. Углубления были гладкие и ровные, хорошо отшлифованные.

"Для чего они предназначались? - подумал Эрик. Не слишком ли они хороши для этого времени?..." Тут же он выругал себя. Как же он мог забыть о своей миссии! Не время стоять и рассматривать камни!

Он взглянул на детектор и вздрогнул от неожиданности. Стрелка крутилась, как бешенная. Он нашел источник временной аномалии.

7. (Италия. 5048 год до н.э.)

Позади Эрика послышался шорох. Он обернулся, рука его легла на приклад его единственного оружия - камнемета. Оружие было изобретено по специальному заказу в физической лаборатории Института Времени. Оно не требовало специальных пуль или других снарядов и могло быть заряжено практически чем угодно - мелким камешком, кусочком металла или дерева. Мгновенно нагревая снаряд до температуры в несколько сот градусов, оружие вышвыривало его из широкого раструба, которым кончалось дуло.

Камнемет не был заряжен. Эрик наклонился и собрал с земли горсть камешков. Еще раз взглянув в направлении, откуда доносился шум, он был поражен тем, что увидел.

Из темноты, прямо в его сторону, шел скелет. Медленно, словно боясь рассыпаться на груды костей, переставляя ноги, он шагал к Эрику. В руках скелет держал длинный меч и деревянных щит, окованный полосками металла.

Чувствуя, как холодеет его затылок, Эрик вложил камень в магазин оружия и выстрелил. Раскаленный снаряд понесся к скелету, волоча за собой огненный шлейф и оставляя дымный след. Он попал в позвоночник скелета и разлетелся во все стороны горячими осколками. Скелет отбросило назад, он упал на спину и распался на отдельные кости. Меч звякнул о каменную плиту. Череп откатился в сторону и лежал, скалясь на Эрика. В глазах его медленно угасали красные угольки ярости... Эрик подобрал череп и сунул его в свой рюкзак. "Профессор разберется, - подумал он. - А пока - вперед. Надо понять, что же произошло со временем".

Осторожно Эрик двинулся в ту сторону, откуда появился скелет. Прямой полутемный туннель завел его в тупик. Но в этом тупике был предмет, который явно не принадлежал к этой эпохе!

Поддерживаемый в воздухе непонятным силовым полем, перед Эриком висело нечто, напоминающее хрустальный шар, наполненный белесым туманом. Эрик выстрелил в него - поле поглотило заряд. Второй, третий - тоже. Четвертый раскаленный камешек, наконец, пробил защиту, и шар со звоном упал на каменный пол. Эрик поднял его - на шара не было ни одной царапины и он был теплым.

Спереди послышался шорох - Эрик поднял глаза и всмотрелся в темноту. У дальней стены он заметил статую, изображавшую какого-то древнего монстра. Глаза статуи горели.

8. (Италия. 5048 год до н.э.)

Следующие несколько часов - а может быть дней или недель были для Эрика постоянным кошмаром. Чудовищные, гротескные существа населяли подземный лабиринт, куда он попал волею судеб...

Хрустальные шары, пять тяжелых сфер, наполненных туманом, оттягивали его плечи. Они лежали в рюкзаке вместе с запасом мелкой гальки для камнемета и еще несколькими странными вещами. Эрик устал и едва передвигал ноги. Найти выход из подземелья он так и не смог.

"На этом и завершена моя миссия, - подумал Эрик.- Причину аномалии я так и не выяснил, исправить ничего не смог, да еще по глупости похоронил себя заживо в этом склепе..."

В этот момент Эрик споткнулся и упал, больно ударившись коленом. Встав, он увидел в каменной плите пола пять сферических углублений.

Бесцельные блуждания в освещенных призрачным светом подземельях вновь привели его к источнику темпоральной энергии.

Некоторое время Эрик стоял, молча, разглядывая углубления. Потом развязал рюкзак и вынул шары. Один за одним он выкладывал их в углубления, и каждый, заняв свое место, начинал светиться изнутри... Уложив пятый шар в центр креста, Эрик вынул из кармана еще одну вещь.

Это были пять металлических полусфер, соединенные тонкими перемычками в форме креста - четыре по краям, одна в центре...

Эрик осторожно накрыл шары этим предметом. В этот момент он почти физически ощутил, как приостановилось на мгновение время, а затем рванулось внезапно, как раскручивающаяся пружина... Мощный темпоральный толчок отбросил его за пределы этого времени...

9. (Нью-Йорк. 2026 год н.э.)

Высокий молодой человек шел ранним утром по коридорам Института Времени. Здание было незнакомо ему - он часто обращался к проходившим мимо него людям с просьбой указать направление. Наконец он достиг своей цели. Дверь кабинета была приоткрыта. Человек громко постучал и заглянул в кабинет.

- Могу я видеть профессора Гаспара Адамсона?

- Входите, молодой человек. Я - Адамсон. Чем могу служить? - профессор привстал из-за стола, отложив в сторону газету.

- Я пришел по просьбе одного человека. Его зовут Эрик. Он просил передать Вам, что вернулся и хочет Вас видеть.

На белоснежных простынях в чистой и светлой комнате под тонким белым одеялом лежал человек. Руки его, покоившиеся поверх одеяла, были забинтованы до самых локтей. На голове также была повязка. Верхняя губа была слегка вздернута неровно заросшим шрамом. Глаза человека были устремлены в потолок, грудь равномерно вздымалась.

Тихо скрипнула дверь, и на кафельный пол комнаты вступили двое в белых халатах.

-Постарайтесь говорить поменьше, не утомляйте его - сказал один. Второй кивнул ему и тот вышел. Оставшийся подошел к кровати и тихо позвал:

-Эрик!...Эрик!

Звук голоса вывел лежавшего из задумчивого созерцания потолка, и он медленно повернул голову.

-Это Вы профессор? Добрый день... Я... Я выполнил миссию и вернулся.

-Да, да... Чуть попозже Вы расскажете мне все подробно... А сейчас Вам необходим покой и сон...

Позже, когда выздоровеете...

Лежавший с трудом кивнул и закрыл глаза. Некоторое время он молчал, затем опять открыл глаза и взглянул на профессора.

-Это был Дамерон, Ангел Хаоса. Я преследовал его в четырех эпохах, исправляя то, что он сделал. Из Италии меня вышвырнуло в Грецию, оттуда - в Древнюю Скандинавию, а когда я все исправил и в том времени - в Египет. Это была последняя точка искривленного времени, и когда спираль вошла в норму, ме-

ня выкинуло куда-то, в место, которое было вне времени и пространства... - Эрик опять закрыл глаза и глубоко вздохнул. Перед ним вновь вставали ужасы пережитого. - Там был Он. Это было Его место, и Он пытался уничтожить меня. Но Его сила уже ушла, я разбил все, что Он сделал - и я победил. Течение времени вновь вынесло меня в мою эпоху, правда, на двенадцать лет позже... Вот и все, профессор...

Через минуту он уже был во власти глубокого сна без сновидений, а профессор Гаспар Адамсон тихо покинул палату госпиталя...

Игра NETHER EARTH уже неоднократно освещалась на страницах нашего журнала. Публикуемый ниже материал - несколько иного рода. Эта компьютерная новелла, которую прислал нам Сергей Татаринов из Калужской обл. Итак,

NETHER EARTH

"...В следующее мгновение он с шипением распоротого воздуха сделал одно сложное и длинное движение, состоящее из перетекающих друг в друга положений меча, и Такэде показалось, что Сухов оплетен, мечем со всех сторон..." Именно в этот момент и сработал чрезвычайный вызов на моем персональном браслете. С сожалением отложив "Посланника", я подошел к визофону.

Из вспыхнувшего экрана на меня взглянуло хмурое лицо шефа нашего отряда "STAR RANGERS" Сержа Полонси:

- Привет, Ник... Похоже, отпуск твой подошел к концу у нас крупные проблемы.

- Судя по твоему лицу, это действительно так.

- Да. Я сейчас собираю всех, до кого могу достать, но почти никого нет на месте работы, как всегда, по уши. Инсигнианский десантный крейсер обошел наши дальние базы и проник в Систему. Сейчас он целится на Юпитер, скорее всего, идет к Европе на захват заводов "Юникс Корпорэйшн". Ты представляешь, что будет с нашей обороной на планетах, если им это удастся? Твоя задача продержаться, пока я смогу собрать хотя бы одну эскадрилью "Гепардов". Не думаю, что крейсер вяжется в бой, это не в их привычках. Скорее, как всегда, высадит десант и смоеется.

- Ладно, понял. Когда вылетать?

Полчаса назад. Черт бы побрал эту "Юникс" все заводы автоматические кто заплатит, на того и будут работать. В общем, на месте разберешься. Мы ведь работали там полгода назад... К сожалению, наскребли всего 20 космокредов, но будем пересылать все, что сможем высвободить. Двигай, рейнджер!

Ну и что мне оставалось делать? Правильно, через полчаса я стартовал на "Гепарде" в сторону Юпитера. Жалко было улетать, я набрал кучу литературы, так мечтал поваляться на диване, забыть про все эти погони и стрельбу. Хотел отдохнуть в первый раз за два года противостояния Содружества и Инсигниана, огромной галактической крепости, появившейся на дальних подступах к Солнечной системе из глублин космоса.

В переговоры Инсигнианцы не вступали; напав на горноперерабатывающие заводы в системе Урана, они уничтожили весь обслуживающий персонал, не оставив в живых никого. Ясно, что крепости требовалось огромное количество разнообразных материалов. Поэтому инсигнианцы стремились сохранить заводы.

К огромному сожалению, Содружество было не в состоянии покончить с захватчиками и, в свою очередь, не могло обходиться без этих материалов. Поэтому заводы переходили из рук в руки с переменным успехом, пока ученые пытались разработать что-либо действенное.

Хотя, конечно, "Юникс Корпорэйшн" свое производство могло расположить и поближе, но, с другой стороны, свихнувшийся экспериментальный робот "Немезис" разнес в пыль три комплекса лазерной связи и город-матку со всеми лабораториями в Море Дождей на Луне, всего лишь из-за короткого замыкания в стратегическом блоке. Разбить его удалось с орбиты только через неделю, когда он добрался до региональной гелиостанции.

С тех пор все заводы по изготовлению и сборке боевых роботов действовали только в Дальнем Внеземье.

...Через три дня на переднем экране всплыл огромный водородный пузырь бурого Джупа. На курсографе "Гепарда" выползла карта-схема его окрестностей, и, нажав клавиши, я вывел на монитор данные подхода к Европе.

Задав курс, я занялся подготовкой посадочного модуля. Заверещавший звонок заставил меня повернуться к пульту локатора. Прекрасный пейзаж разлетающиеся осколки охранных спутников и инсигнианский крейсер, не обращая на меня внимания, пикирует к поверхности Европы! Нападать бесполезно, огневая мощь и скорость у меня не такие, чтобы с ним тягаться, да и он не будет со мной связываться пока я

доберусь до орбиты, он свое дело уже сделает. Проклятые рептилии всегда действовали одинаково высадив десант и подавив сопротивление, корабли немедленно стартовали к Инсигниану за подкреплениями. Обычно такая тактика себя оправдывала, но сейчас это был мой шанс.

Определив местонахождение одной из баз, я послал вниз модулированный частотный сигнал. База ответила значит, пока не захвачена.

Плавно опустив модуль на посадочную площадку, я выдвинул штангу манипулятора и подключился к главной контрольной цепи управления.

Ну что ж! Инсигнианцы уже захватили три базы и успели выпустить двух роботов. Они знают и обо мне через цепи своих баз, так что сейчас все решает скорость. Но ведь им еще нужны и заводы, поэтому они будут отвлекаться на захват, а моя задача совсем в другом! Хорошо, что поверхность спутника не везде легкопроходима, да и остатков всяких построек тут хватает, так что и это мне на руку.

Быстренько отстучав запрос на компьютере и переслав в банк базы все свои кредиты, я с удовлетворением увидел, как из бронированных ворот легко выплыл на антигравях огромный полнокомплектный робот серии БР-4. Не хватало только блока наведения и радара. Ну, это мы и сами сможем!

Аккуратно зафиксировав модуль на "голове" робота магнитными присосками, я проверил управление. Прекрасно полный комплект ракет и снарядов для двух скорострельных пушек! А опробовав фазор на подвернувшейся скале, я пришел в восторг такая мощь!

Ну, двигаем, а то роботов у инсигнианцев уже пять штук, да и один завод они захватили.

Робот с легким гулом скользил над гладкими ледовыми пустошами и острыми скалами Европы. На обзорном экране проплывали корпуса заводов. Помимо заводов попадались высокие полуразрушенные стены каких-то строений. "Если удастся использовать их для укрытий..." мелькнула мысль, и тут на радаре, среди лабиринта стен и заводов, высветилось несколько точек. "Они!" Один на моих глазах заехал во двор завода. "А скорость-то у них поменьше моей" пронеслось в голове, руки в это время порхали по клавишам, запуская боевую программу.

Судя по передвижению роботов, стало ясно, что действуют они без "хозяев", по программе. Перемещались они слишком бестолково, да и логика военных компьютеров никогда не была на высоте. К тому же все вооружение смонтировано на биподах. На мое счастье.

Выдвинув из-за стенки ствол фазора, я дождался, пока противник подставит свой борт, и нажал на гашетку. Трех фиолетовых молний было достаточно, чтобы робот осел грудой чадающего металла. Другой поспешил занять его место, но, застряв в обломках, не успел развернуться. Груда металлолома увеличилась.

Судя по часам, заканчивался второй день моего пребывания в Европе. Я успел уничтожить двух роботов, не получив ни одной царапины, и был на подступах ко второй базе. Расстреляв ракетами издалека еще одного робота, я направил свою "лошадку" к базе, ослабив внимание. За что и был тут же наказан двумя ракетами из распахнувшихся ворот прямо в лоб.

"Впредь тебе наука!" думал я со злостью, ответным залпом фазора и ракет уничтожая появившегося противника. На мониторе вспыхивала красная надпись "Уничтожен первый слой защиты. Боеготовность снижена на 15%!"

Из-за построек базы выдвинулась ракетно-артиллерийская платформа и начала разворачиваться в мою сторону. Когда и этот противник, развалившись надвое, осел на обломки скал, я вытер пот: "Надо менять тактику так они меня быстро в порошок сотрут!"

Запустив в мозг робота программу поиска и ликвидации роботов противника, я освободил его и подключился к центру управления базой.

Приятная неожиданность! Серж сдержал обещание, на моем счету скопилась сумма, позволяющая собрать еще одного робота. Инсигнианский крейсер превращен в пыль за орбитой Урана, не успев подать сигнал о помощи. В мою сторону двигалась эскадрилья "RANGERS"; дней через пять они должны были быть здесь.

Ну, уж нет! Раз помощи этим крокодилам ждать неоткуда, тут я их и похороню!

Я повернулся к радару, на экране вспыхнула и пропала красная точка, мой подопечный только что превратил противника в мусор. Быстренько набрав заказ и перелетев на робота, я подключился к управлению...

Применяя новую тактику и обходя дымящиеся металлические завалы, я упорно продвигался вперед. Через девять дней, полностью вымотанный непрерывным сражением, я наблюдал, как перед моим "Гепардом" возник строй серебристых вертких машин. На этот раз лицо Сержа на экране было веселым:

Ник, разбойник, сколько железа ты наломал, дорвался до стрельбы!!!

Динамики трещали от веселых поздравлений, истребители один за другим "проваливались" к спутнику, спеша осмотреть картину боев. Перед моим мысленным взглядом проплывали картины рассыпаю-

щихся роботов, мелькали фиолетовые молнии фазоров и оранжевые вспышки ракет, в ушах слышался так надоевший гул и скрежет, продолжавшийся все девять дней.

- Все, все, все - теперь только отдыхать и читать... - подумал я, вводя в курсограф обратный курс. На пульте коммутатора вспыхнула ярко-красная лампочка, и взвизгнул сигнал: "Боевая тревога! Всем рейнджерам, всем подразделениям флота! В квадрате U-4 замечена эскадра противника. Курс предположительно на Титанию! Повторяю..."

TRICKS ZX (КОДЫ, ПАРОЛИ, ХИТРОСТИ)

CONFIDENTIAL

Сегодня в этом разделе мы приведем несколько СПЕЦЗАГРУЗЧИКОВ для фанатов игровых программ.

BALLBREAKER

```
10 PAPER 0
20 CLEAR 28000: LET S=0: LET T=0
30 FOR I=23296 TO 23334: READ A: POKE I, A: LET T=T+A: NEXT I
40 IF T<>4868 THEN PRINT "ERROR IN DATA": STOP
50 FOR I=1 TO 1E9: READ A
60 IF A<256 THEN POKE I, A: NEXT I
70 RANDOMIZE USR 23296
80 DATA 243, 221, 33, 172, 253, 17, 63
90 DATA 4, 62, 255, 55, 205, 86, 5, 48
100 DATA 241, 205, 86, 5, 49, 0, 0, 221, 33
110 DATA 214, 109, 17, 237, 134, 175, 55, 205
120 DATA 227, 254, 48, 242, 175
130 DATA 50, 164, 155: REM INFINITE MISSILES
140 DATA 50, 0, 148, 50, 64, 148: REM INFINITE BALLS
150 DATA 50, 101, 139, 62, S, 50, 112, 128: REM NO FLASHY START SCREEN
160 DATA 195, 0, 128, 999: REM END MARKER
```

BASIL, THE GREAT MOUSE DETECTIVE

```
10 CLEAR 27E3: LOAD "" CODE
20 POKE 55837, 251
30 POKE 55838, 201
40 RANDOMIZE USR 3E4
50 POKE 41296, 0: RUN USR 32768
```

BATTY

```
10 CLEAR 60000: LOAD ""CODE: POKE 65522, 252
20 FOR N=64512 TO 64517: READ A: NEXT N
30 PRINT USR 64753
40 DATA 50, 56, 189, 195, 0, 104
```

```
10 CLEAR 6E4: LET t=0
20 FOR f=64512 TO 64616
30 READ a: POKE f, a
40 LET t=t+(f-64502)*a : NEXT f
50 IF t<>766540 THEN STOP
60 LOAD "" CODE
70 POKE 65522, 252
80 RANDOMIZE USR 64753
90 DATA 33, 44, 252, 17, 0
100 DATA 91, 1, 200, 0, 237
110 DATA 176, 62, 205, 50, 236
120 DATA 161, 50, 167, 151, 33
130 DATA 0, 91, 34, 168, 151
140 DATA 33, 0, 0, 34, 170
150 DATA 151, 33, 9, 91, 34
160 DATA 237, 161, 175, 50, 239
170 DATA 161, 195, 0, 104, 205
180 DATA 17, 91, 219, 254, 17
190 DATA 230, 31, 201, 205, 17
200 DATA 91, 219, 31, 230, 31
210 DATA.201, 229, 197, 213, 245
```

```
220 DATA 205, 142, 2, 123, 254
230 DATA 37, 40, 13, 254, 29
240 DATA 40, 12, 254, 21, 40
250 DATA 12, 241, 209, 193, 225
260 DATA 201, 175, 24, 11, 62
270 DATA 183, 24, 7, 62, 4
280 DATA 50, 232, 183, 24, 237
290 DATA 50, 56, 189, 24, 232
```

BAZOOKA BILL

```
10 BORDER 0: PAPER 0: INK 7: CLS
20 LOAD "" SCREEN$
30 LOAD "" CODE 37088
40 LOAD "" CODE 24800
50 POKE 28489, 201
60 LET L=USR 24800
```

BEAMRIDER

```
10 CLEAR 27999: LOAD "" CODE 16384
20 POKE 23303, 190
30 FOR a=23350 TO 23356: READ b: POKE a, b: NEXT a
40 LET c=USR 23296
50 DATA 175, 50, 65, 199, 195, 80, 195
```

BLACK MAGIC

```
10 CLEAR 3e4: LET t=0
20 FOR f=23296 TO 23403
30 READ a: POKE f, a
40 LET t=t+(f-23286)*a: NEXT f
50 IF t<>572714 THEN STOP
60 LET t=0: FOR f=f TO 1e9
70 READ a
80 IF a>255 THEN GO TO 110
90 POKE f, a
100 LET t=t+(f-23394)*a : NEXT f
110 IF t<>a THEN STOP
120 LOAD "" CODE: RUN USR 23296
130 DATA 42, 110, 91, 237, 123
140 DATA 108, 91, 221, 33, 102
150 DATA 91, 6, 6, 197, 6
160 DATA 0, 221, 78, 0, 221
170 DATA 35, 17, 72, 88, 229
180 DATA 197, 237, 176, 33, 75
190 DATA 88, 126, 254, 54, 32
200 DATA 2, 54, 0, 235, 54
210 DATA 201, 30, 150, 205, 72
220 DATA 88, 193, 225, 9, 193
230 DATA 16, 217, 42, 112, 91
240 DATA 237, 91, 114, 91, 1
250 DATA 0, 4, 237, 176, 42
260 DATA 116, 91, 237, 91, 118
280 DATA 91, 1, 0, 4, 237
290 DATA 176, 42, 120, 91, 54
300 DATA 0, 42, 122, 91, 54
310 DATA 126, 35, 54, 91, 42
320 DATA 124, 91, 233, 1, 96
330 DATA 54, 82, 50, 17, 14
340 DATA 12, 24, 16, 24, 50
350 DATA 14, 25, 22
360 DATA 200, 175, 192, 228, 107, 238, 4, 252, 194, 59, 127, 82, 114, 255,
66, 255, 6, 255, 175, 50, 154, 96, 49, 255, 255, 195, 0, 91, 93648
```

BOBBY BEARING

```
10 CLEAR 65000
20 LET T=0
30 FOR N=64223 TO 64316: READ A
40 POKE N, A: LET T=T+A: NEXT N
50 IF T<>12120 THEN PRINT "DATA ERROR IN MAIN BLOCK"
60 FOR N=N TO 1E9: READ A: IF A<256 THEN POKE N, A: LET T=T+A: NEXT N
70 IF T<>A THEN PRINT "DATA ERROR IN GAME BLOCK": STOP
80 RANDOMIZE USR 64225
90 DATA 24, 46, 33, 89, 5, 93, 22, 250, 1, 134, 0, 213
100 DATA 237, 176, 235, 35, 35, 34, 95, 250, 46, 122
110 DATA 112, 46, 215, 116, 46, 207, 54, 188, 33, 38
120 DATA 210, 34, 126, 250, 225, 49, 61, 251, 221, 225
130 DATA 209, 191, 55, 8, 249, 233, 49, 65, 251, 221
140 DATA 225, 209, 59, 225, 193, 205, 29, 251, 24, 43
150 DATA 221, 110, 0, 124, 170, 171, 221, 172, 221
160 DATA 173, 173, 221, 119, 0, 124, 203, 99, 40, 3
170 DATA 129, 131, 146, 128, 103, 221, 35, 27, 122
180 DATA 179, 32, 225, 201
200 DATA 186, 64, 183, 64, 183, 181, 0, 91, 111, 155, 16, 106, 110
210 DATA 62, 36, 50, 190, 109, 195, 0, 91, 14056
```

BOMB JACK

```
10 CLEAR 29877: FOR N=23371 TO 23377: READ A: POKE N, A: NEXT N
20 DATA 175, 50, 92, 170, 124, 246, 1, 201
30 LOAD "" CODE
40 POKE 64909, 195: POKE 64910, 232: POKE 64911, 253
50 RANDOMIZE USR 64767
```

BOMB JACK II

```
10 CLEAR 60000: LOAD "" CODE: POKE 65226, 250
20 FOR N=64000 TO 64006
30 READ A: POKE N, A: NEXT N
40 RANDOMIZE USR 64739
50 DATA 175, 50, 35, 99, 195, 0, 91
```

BOMBSCARE

```
10 BORDER 0: PAPER 0: INK 0
20 CLEAR 27999: POKE 23659, 0
30 LOAD "" SCREEN$: PRINT AT 21, 0: LOAD ""CODE 28000
35 PRINT AT 21, 0: LOAD "" CODE 65000
40 POKE 23606, 46: POKE 23607, 181: POKE 23659, 2
50 POKE 56777, 0: REM INFINITE LIVES
60 POKE 54129, 0: REM INFINITE SHOTS
70 POKE 56286, 0: REM INFINITE ENERGY
80 POKE 57316, 201: REM INFINITE TIME
90 RANDOMIZE USR 65000
```

BOOTY

```
10 CLEAR 26000: LOAD "" CODE: LOAD "" CODE 26880: RANDOMIZE USR 26880: POKE
58294, 0: RANDOMIZE USR 52500
```

BOSCONIAN

```
10 CLEAR 25599: LOAD "" CODE 16384
20 LOAD "" CODE 25600: LOAD "" CODE 32765
30 POKE 33356, 60: RANDOMIZE USR 32765
```

BOULDERDASH

```
10 POKE 23693, 4: CLEAR 24100: LOAD "" CODE
20 LOAD "" CODE: RANDOMIZE USR 65100: LOAD "" CODE
30 POKE 28929, 8: POKE 28934, 8: POKE 28939, 8
40 RANDOMIZE USR 41712
```

BOUNDER

```
10 LET CS=0: FOR N=23296 TO 23301
```

```
20 READ A: LET CS=CS+A
30 POKE N, A: NEXT N
40 IF CS<>464 THEN PRINT "ERROR": STOP
50 LOAD ""
100 DATA 66, 79, 85, 178, 67, 89
```

BRIDE OF FRANKENSTEIN

```
10 CLEAR 3e4: LET t=0
20 FOR f=23296 TO 23403
30 READ a: POKE f, a
40 LET t=t+(f-23286)*a:NEXT f
50 IF t<>572714 THEN STOP
60 LET t=0: FOR f=f TO 1e9
70 READ a
80 IF a>255 THEN GO TO 110
90 POKE f, a
100 LET t=t+(f-23394)*a: NEXT f
110 IF t<>a THEN STOP
120 LOAD ""CODE: RUN USR 23296
130 DATA 42, 110, 91, 237, 123
140 DATA 108, 91, 221, 33, 102
150 DATA 91, 6, 6, 197, 6
160 DATA 0, 221, 78, 0, 221
170 DATA 35, 17, 72, 88, 229
180 DATA 197, 237, 176, 33, 75
190 DATA 88, 126, 254, 54, 32
200 DATA 2, 54, 0, 235, 54
210 DATA 201, 30, 150, 205, 72
220 DATA 88, 193, 225, 9, 193
230 DATA 16, 217, 42, 112, 91
240 DATA 237, 91, 114, 91, 1
250 DATA 0, 4, 237, 176, 42
260 DATA 116, 91, 237, 91, 118
280 DATA 91, 1, 0, 4, 237
290 DATA 176, 42, 120, 91, 54
300 DATA 0, 42, 122, 91, 54
310 DATA 126, 35, 54, 91, 42
320 DATA 124, 91, 233, 1, 96
330 DATA 54, 82, 50, 17, 14
340 DATA 12, 24, 16, 24, 50
350 DATA 14, 25, 22
360 DATA 255, 183, 237, 228, 86, 238, 239, 249, 225, 253, 185, 85, 90, 253,
42, 253, 241, 252, 62, 201, 50, 28, 158, 49, 119, 105, 195, 144, 133, 107475
```

BRUCE LEE

```
10 LET TOT=0: FOR N=50000 TO 50084: READ A: LET TOT=TOT+A: POKE N, A: NEXT N
20 IF TOT<>10376 THEN PRINT AT 10, 10;"ERROR IN DATA": STOP
30 PRINT AT 10, 10;"INSERT TAPE": LOAD "" CODE: RANDOMIZE USR 50000
100 DATA 62, 255, 55, 17, 141, 5, 221, 33
101 DATA 203, 92, 205, 86, 5, 48, 241, 33
102 DATA 86, 176, 34, 233, 96, 58, 202, 96
103 DATA 238, 194, 50, 202, 96, 33, 147, 195
104 DATA 17, 28, 238, 1, 32, 0, 237, 176
105 DATA 243, 237, 94, 33, 41, 236, 229, 235
106 DATA 33, 137, 97, 229, 51, 51, 1, 242
107 DATA 1, 33, 253, 94, 62, 200, 237, 79
108 DATA 195, 137, 97, 62, 195, 50, 80, 202
109 DATA 62, 96, 50, 81, 202, 62, 201, 50
110 DATA 200, 218, 195, 0, 226
```

BUBBLE BOBBLE

```
10 LET T=0
20 FOR F=32768 TO 32876
30 READ A: POKE F, A
40 LET T=T+(F-32758)*A: NEXT F
50 IF T-679460 THEN STOP
60 FOR F=32877 TO 1E9: READ A
70 IF A>256 THEN GO TO 100
80 POKE F, A
90 LET T=T+(F-32867)*A: NEXT F
100 IF T-A THEN STOP
102 POKE 32837, 90: POKE 32778, 90: POKE 32843, 91: POKE 32784, 91
110 LOAD "" CODE: RUN USR 32830
120 DATA 50, 21, 255, 122, 254
130 DATA 46, 192, 62, 50, 50
140 DATA 84, 205, 33, 21, 255
150 DATA 34, 85, 205, 62, 195
160 DATA 50, 58, 0, 33, 32
170 DATA 128, 34, 59, 0, 195
180 DATA 0, 0, 50, 107, 92
190 DATA 33, 114, 128, 17, 0
200 DATA 0, 1, 0, 0, 237
210 DATA 176, 195, 61, 0, 128
220 DATA 223, 181, 209, 177, 144
230 DATA 141, 139, 151, 206, 198
240 DATA 199, 200, 237, 123, 112
250 DATA 128, 62, 205, 50, 84
260 DATA 205, 33, 0, 128, 34
270 DATA 85, 205, 33, 109, 128
280 DATA 126, 50, 22, 128, 50
290 DATA 28, 128, 50, 31, 128
300 DATA 50, 40, 128, 50, 48
310 DATA 128, 35, 126, 50, 42
320 DATA 128, 35, 126, 50, 39
330 DATA 128, 195, 0, 205
340 DATA 91, 9, 213, 80, 195, 62, 52, 50, 95, 171, 195, 188, 245, 707604
```

BUBBLER

```
10 CLEAR 24064: LOAD ""CODE
20 RANDOMIZE USR 32768
30 LOAD "" CODE: POKE 57517, 0
40 RANDOMIZE USR 61440
```

BUTCH HARD GUY

```
10 LET T=0: FOR F=3E4 TO 30043
20 READ A: POKE F, A
30 LET T=T+(F-29990)*A:NEXT F
40 READ A: IF T<>A THEN STOP
50 RANDOMIZE USR 3E4
60 DATA 62, 255, 50, 203, 92
70 DATA 221, 33, 0, 64, 17
80 DATA 0, 28, 62, 255, 55
90 DATA 205, 86, 5, 48, 241
100 DATA 33, 82, 117, 17, 19
110 DATA 91, 1, 11, 0, 237
120 DATA 176, 195, 0, 91, 175
130 DATA 50, 64, 138, 50, 72
140 DATA 92, 195, 133, 158, 133681
```

ГДЕ МОЖНО ПРИОБРЕСТИ НАШИ ИЗДАНИЯ.

1. Торговый центр фирм "ИНФОРКОМ" и "ФОРМАК".
 - г. Москва, Новый Арбат, д.2. 19-е отделение связи. 1-й этаж операционного зала. Кроме воскресенья, с 10 до 14 и с 15 до 17 ч.
 - Розничная и оптовая продажа по издательским ценам. Подписка на ZX РЕВЮ. Авторские и фирменные программы по каталогу.
2. г. Чебоксары, ул. Ярославская, д. 64, офис 24. Предприятие. "Мой город".
3. г. Ростов-на-Дону, ул.Кручинина, Ростовский радиорынок. Магазин фирмы "СКОРПИОН". В будние дни с 9 до 15 ч., в вых. с 8 до 13 ч.
4. г. Йошкар Ола, пл. Ленина,3. Гл. корпус ПОЛИТЕХа "МОДУЛЬ".
5. г. Санкт-Петербург. Радиорынок в Автово. Палатка № 149-"SCORPION".
6. г. Ангарск, Ворошилова, 10. "SPECTRUM".
7. г. Иркутск, ул.Урицкого, 2. "Все для ZX SPECTRUM".
8. г. Одесса, радиорынок, под вывеской "AcashiSoft" или по тел.33-71-71.
9. г. Барнаул, ул. Э.Алексеевой, 2-а. тт. 76-42-98, 76-42-25. Фирма КОБЭЛ.

ВНИМАНИЮ ОПТОВЫХ ПОКУПАТЕЛЕЙ!!!

**Адреса пунктов продажи изданий МКП «ИНФОРКОМ» и
ТОО «ФОРМАК» публикуются БЕСПЛАТНО.**

**Для Вас разработана гибкая система скидок на полиграфическую
продукцию (от 10% до 40%), осуществляется поставка по
почте, информационная и рекламная поддержка**