



"ИНФОРКОМ" 121019, Москва, Г-19, а/я 16

СПЕКТРУМ в школе

Здравствуйтесь, дорогие друзья!

Так уж повелось в последнее время на наших страницах, предназначенных для школьников, что мы играючи осваиваем интересные численные методы решения математических уравнений, и сегодня мы хотели бы предложить Вам еще одну игру, из которой вытекают совсем не игровые последствия.

Скажите, что Вы подумаете о человеке, который предложит Вам решить ОДНО уравнение с двумя неизвестными? Наверное, что у него, как сейчас говорят, "крыша поехала". А если это будет одно уравнение с пятью неизвестными? Это невозможно, скажете Вы? А если он Вам скажет, что его не интересует абсолютно точное решение и его устроит решение с какой-то допустимой погрешностью? О, это совсем другое дело - тогда можно как-то попробовать УГАДАТЬ более-менее приличное решение, особенно если это игра.

Итак, возьмем для игры следующие значения:

$$X = 5; \quad Y = 2;$$

Составим из них уравнение, например такое:

$$X + X*Y + X*X + Y*Y - 44 = 0$$

И предложим его решить кому-либо из знакомых, не открывая истинных значений X и Y. Единственное, что можно подсказать - так это то, что X находится в пределах от 0 до 20, а Y - от 0 до 10, и пусть он теперь попробует решить такое уравнение.

Что будет делать Ваш партнер? Во-первых, он попробует взять в качестве первого приближения средние значения X и Y из их возможного интервала и посмотреть, что из этого получится:

$$X = 10; \quad Y = 5$$

$$10 + 50 + 100 + 25 - 44 = 139$$

Мы получили "невязку, равную 139, поскольку не угадали X и Y.

Надо их корректировать. Поскольку "невязка" больше нуля, значит мы где-то перебрали. Ударим опять посередине:

$$X = 5; \quad Y = 2.5.$$

Получим:

$$5 + 12.5 + 25 + 6.25 - 44 = 4.75$$

"Невязка" стала значительно меньше, значит, мы на верном пути. Попробуем еще раз уменьшить X и Y.

$$X = 2.5; \quad Y = 1.25$$

$$2.5 + 3.13 + 6.25 + 1.56 - 44 = -30.56$$

На сей раз мы ушли не в ту сторону. Попробуем еще раз и опять посередине:

$$X = 3.75; \quad Y = 1.85$$

$$3.75 + 6.98 + 14.06 + 6.98 - 44 = -12.23$$

Снова "недолет", стрельнем еще последний разок и пока хватит.

$$X=4.38; \quad Y=2.18$$

$$4.38+9.55+19.18+4.75-44 = -6.14$$

Итак, мы получили результат:

$$X=4.38, \text{ а } Y=2.18$$

Истинный же результат, как мы помним, был:

$$X=5, \text{ а } Y=2.$$

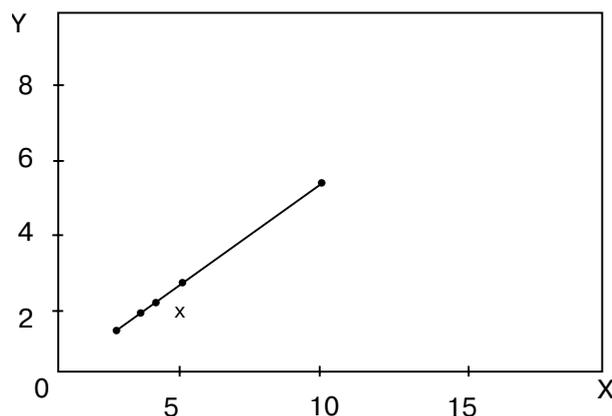
Выходит, мы решили задачу с точностью до примерно 12 процентов. Хорошо это или плохо? Ну, во-первых, это еще не предельная наша точность, ведь считали-то мы на бумажке, а если применить компьютер, то ее можно бы и повысить и как это сделать, мы еще покажем. А во-вторых, надо подумать, где такие расчеты могут использоваться?

Оказывается, они могут применяться, как правило, в двух случаях. Во-первых, в технике, когда какие-то параметры конструкции неизвестны или известны плохо. Их, конечно, можно устанавливать с помощью испытаний, экспериментов и т.п. и так и делают, но это довольно дорого, трудоемко и долго. Такой расчет позволяет во много раз снизить затраты на испытания и сэкономить месяцы и годы. К тому же, как правило, все технические устройства имеют не менее, чем 100-процентный запас надежности и точность наших расчетов вовсе не так уж и плоха.

Вторая область, когда приходится решать уравнения, в которых неизвестных больше, чем самих уравнений - экономические расчеты, типа того, как организовать транспортировку пива, нефти и подсолнечного масла между десятью регионами, спрос в которых на эти продукты точно не известен, но затратить при этом минимум средств на перевозку, хранение и обслуживание цистерн. В этих расчетах точность порядка 12 процентов - просто блестящая. В рыночной экономике надо уметь прогнозировать результаты своих действий.

Итак, мы умеем теперь решать уравнения с большим количеством неизвестных, хотя вопросов еще осталось немало. Во-первых, как приспособить к этому делу компьютер, а во-вторых, надо ему дать четкий алгоритм, потому что мы с Вами "стреляли" по собственному разумению - чуть ближе, чуть дальше, а компьютеру надо точно объяснить, куда "стрелять" в поисках нужного решения.

Давайте посмотрим на графике (рис. 1), как мы двигались к решению, начиная от $X=10, Y=5$.



Как видите, мы все время "бегали" по одной прямой, пока не нашли на ней точку, близко лежащую к истинному решению. Да это и не удивительно, ведь мы все время делили отрезок пополам. Но так дело не пойдет. Это хорошо, что у нас только два неизвестных и наш график имеет две координаты X и Y (его можно изобразить на даже на плоскости). А если бы неизвестных было штук пять? Вы не представляете, что такое пятимерное пространство? Мы тоже не представляем, и как попасть прямой линией в таком пространстве в нужную точку? Это действительно трудно представить.

Для того, чтобы "освободить" себя от блуждания по одной прямой и выйти за ее пределы (например так, как показано на Рис. 2), надо "отпустить" X и Y от необходимости делить отрезок пополам при поиске нового приближения и разрешить компьютеру самому

выбирать, какое приращение на X и Y он будет делать на каждом шагу. Можно даже разрешить ему это делать случайным образом. Пусть бегаем где хочет, все равно мы примем только те его шаги, которые ведут к уменьшению "невязки", а все прочие отбросим.

Так наш компьютер "выйдет" за пределы роковой прямой. Если у нас пять неизвестных, то он начнет "блуждать" в пятимерном пространстве, медленно но верно подходя к приемлемому (хоть и не точному) решению.

Такой метод расчета известен в математике под названием "случайного" поиска.

Наметим алгоритм:

1. Ввод уравнения.
2. Ввод предельных значения неизвестных.
3. Выбираем первую точку в качестве средней между предельными значениями неизвестных.
4. Расчет "невязки".
5. С помощью генератора случайных чисел определяем "приращение" для каждой из координат.
6. Получаем новую точку.
7. Проверяем, не выходит ли она за пределы допустимых значений по координатам. Если да, то возвращаемся на п. 5.
8. Рассчитываем "невязку" и сравниваем ее с предыдущей "невязкой". Если она по абсолютной величине больше, чем предыдущая, то такой шаг нам не нужен, возвращаемся на п. 5.
9. Делаем новый шаг в том же "направлении", что и предыдущий случайный шаг. Теперь нам генератор случайных чисел не нужен, не стоит тратить время на случайный поиск направления, раз мы его уже нашли.
10. Проверяем не вышли ли мы за пределы допустимой области. Если да, то возврат на п. 5 (снова включаем случайный поиск).
11. Проверяем значение "невязки". Если она возросла, то шаг отменяется и возвращаемся на п. 5 для поиска нового направления, а если уменьшилась, то возвращаемся на п. 9 и шагаем в том же направлении еще раз.

Можно сделать кое-что еще. Во-первых, мы имеем здесь бесконечный цикл, поэтому надо предусмотреть окончание расчетов и выход из него по достижении какого-то значения "невязки".

Во-вторых, можно ускорить работу, если после удачного шага делать следующий шаг в том же направлении например в два раза длиннее. А после неудачного шага не бежать сразу к случайному определению нового направления, а попробовать шагнуть еще раз туда же, но в десять раз короче.

Можно повысить надежность наших расчетов, запуская одну и ту же программу несколько раз. Поскольку у нас неизвестных больше, чем количество уравнений, то может быть получено много разных пар X и Y , которые удовлетворяют решению задачи. Определив случайным образом несколько таких пар, можно подумать и выбрать среди них оптимальную.

Можно предложить еще множество разных усовершенствований и ухищрений и каждый, кому понадобится использовать метод случайного поиска для решения своей задачи сможет доработать его так, как ему надо.

От себя же добавим, что у метода случайного поиска, хоть он и работает довольно медленно, есть одна интересная особенность. Дело в том, что при применении других численных методов расчет хоть происходит и быстрее, но, встречаются ситуации, когда он заходит в тупик. Математики говорят, что расчёт "попал в яму" или "в овраг", и обычными способами из такого оврага не выбраться. А с помощью случайного поиска - пожалуйста. Поэтому применяют иные, быстроработающие методы, но когда оказывается, что "невязка" начинает уменьшаться слишком медленно, то подключают случайный поиск и он выводит из тупика.

Представьте себе, что на неровную холмистую лужайку бросили мяч. Ваша задача -

закатить его в самую низкую точку этой лужайки (минимальная невязка). Если он попал на холм, то он бодро покатится туда, куда надо (работает быстрый численный метод), но если по пути он закатится в какую-либо ямку, рытвину или овраг - дело плохо, сам он оттуда не выкатится. Его надо "встряхнуть", еще раз подкинуть, чтобы он мог катиться дальше - это и сделает вовремя подключенная процедура случайного поиска.

```

10 REM *** Метод случайного поиска***
20 DEF FN A(X,Y) = X+X*Y+X*X+Y*Y - 44: REM Ваше уравнение.
30 LET xmax = 20: LET xmin=0: LET ymax = 10: LET ymin=0: REM ввод граничных условий.
40 LET x=(xmax+xmin)/2: LET y=(ymax+ymin)/2: REM исходная точка поиска
45 GO SUB 400:REM расчет невязки.
48 LET dif1 = dif: REM запомнили невязку.
49 GO SUB 300: REM печать.
50 LET xstep = (xmax-xmin)/10: LET ystep = (ymax-ymin)/10: REM стандартный шаг поиска.
60 GO SUB 500: REM определение случайного направления.
70 LET x1=x+dx: LET y1=y+dy: REM Поиск новой точки.
80 IF x1 > xmax OR x1 < xmin OR y1 > ymax OR y1 < ymin THEN GO TO 60: REM При выходе за
    границы зоны надо повторить поиск новой точки.
90 LET x=x1: LET y=y1: REM переход к новой точке.
100 GO SUB 400: REM расчет новой невязки.
110 IF ABS(dif) < ABS(dif1) THEN GO TO 130
120 LET x=x-dx: LET y=y-dy: GO TO 60: REM если невязка оказалась больше, чем была, то эта
    точка нам не нужна.
130 LET dif1 = dif: REM запомнили новую невязку.
140 GO SUB 300: REM печать
150 IF ABS(dif1) < 1.0 THEN STOP: REM если достигнута заданная точность, то конец работы.
160 GO TO 70: REM Возврат для следующего шага.
300 PRINT "X= ";x; TAB 10;"Y= ";y; TAB 20;"DIFF= ";dif1
310 RETURN
400 LET dif = FN A(x,y)
410 RETURN
500 LET dx=xstep*(2*RND-1):LET dy=ystep*(2*RND-1)
510 RETURN

```

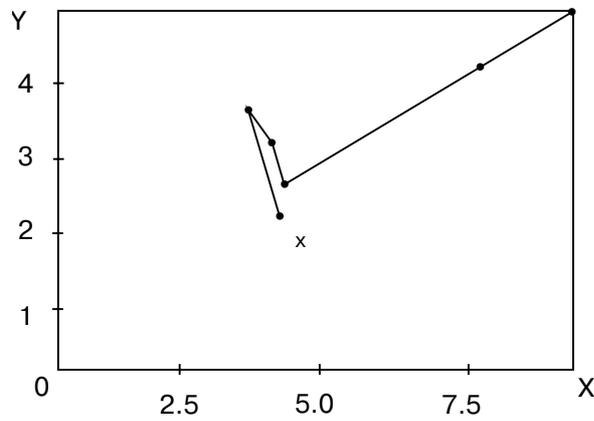
Примечание. Здесь мы используем не функцию RND, а $2 \cdot \text{RND} - 1$. Это сделано потому, что RND дает результат в диапазоне $0 \dots 1$, а нам нужно в диапазоне $-1 \dots +1$, иначе наш поиск будет не случайным, а все время будет уводить нас вправо и вверх.

В строке 50 мы задали параметры xstep и ystep - это как бы сила того толчка, с которой застрявший мяч выбрасывается из ямки. Мы взяли эту величину, равной одной десятой размеров "игрового поля". Если взять мало, то силы может и не хватить и мячик не найдет более удачного продолжения своему пути, а если взять слишком много, то он будет слишком долго метаться по игровому полю, а то и вообще выходить за его пределы. Можно подобрать эту величину экспериментально, а можно и сделать ее плавающей, в зависимости от того, как близко к финалу мы подошли.

Проведя расчеты по приведенной выше программе мы получили следующую траекторию поиска.

N шага	X	Y	Невязка
1	10	5	141
2	6	4.2	78
3	4	2.5	-7.5
4	3.8	3.1	-4.9
5	3.5	3.7	-1.7
6	4.3	2.8	-1.2
7	5.1	2.0	0.66

График поиска показан на рис. 2.



Результат очень близок к задуманному. Но для проверки его точности мы еще пять раз повторили поиск и получили еще пять возможных ответов.

X	Y
1.3	5.8
4.8	2.2
4.0	3.3
2.9	4.4
2.6	4.7

Как видите, здесь результаты уже не столь близки к задуманным, но зато мы можем сузить диапазон "возможных" значения для X и Y. Например так:

XMAX = 6: XMIN = 0:

YMAX = 6: YMIN = 0

И теперь "прогон" программы еще восемь раз дал другую серию результатов:

X	Y
4.55	2.61
4.25	2.98
4.56	2.60
5.21	1.69
4.54	2.61
4.57	2.57
4.58	2.57
4.69	2.43
Среднее 4.65	2.51
Точность 7%	25%

А теперь, в заключение этой статьи, мы предлагаем Вам пару легенд, из которых Вы поймете, что такие "детские игры" могут в реальной жизни приводить к экономии тысяч рублей личных и миллиардов рублей государственных денег.

1.

Дело было почти 20 лет назад. СССР произвел запуск очередной межпланетной станции в дальний космос. Как на грех, сразу после пуска выясняется, что одна из ответственных систем вообще не работает. Причем, пока она и не нужна, а вот когда через несколько месяцев станция дойдет до конечной цели своего путешествия, то без нее не обойтись. Но меры можно предпринимать только сейчас, потом будет поздно. Надо что-то срочно делать.

Было несколько путей выхода из создавшегося положения и окончательный выбор решения лег на плечи молодого специалиста, стаж работы которого был около двух лет. Поставили срок - три дня и поручили обосновать принятое решение необходимыми расчетами.

Уже через под-дня грубых расчетов оказалось, что предсказать, что даст то или иное решение можно только решив пару уравнений размером в пол-страницы, в котором к тому же было пять неизвестных параметров. Конечно, эти параметры можно было "вытаскивать"

ставя практические эксперименты, но на это нужны были бы годы и такая работа оказалась бы на уровне десятка кандидатских диссертаций.

Беглый поиск в технической библиотеке показал, что можно попробовать воевать с таким уравнением. Еще пол-дня ушло на изучение основ вариационного исчисления и теории оптимального управления, а еще через два дня после отсева всего лишнего, почерпнутого из научных трудов, родился алгоритм, немного похожий на предложенный Вам. Решение было найдено и подтверждено расчетами, благо ЭВМ ЕС-1022 в те годы уже существовала.

Дело оставалось за малым - защитить это решение перед Государственной комиссией и убедить ее в правильности предложенных мероприятий.

Начальника отдела чуть не хватил удар, когда он увидел, что решение получено из двух уравнения с кучей неизвестных. Его реакцию словами не передать, и вот тогда и началась игра, которую мы Вам описали в самом начале. Задумывались числа, составлялись уравнения, а молодой специалист "угадывал" то, что задумал начальник. Пришло еще более высокое руководство и увлеченно включилось в игру. Одним словом, вдоволь наигравшись, они вынуждены были перекреститься и согласиться на рискованное с их точки зрения решение. Их поддержка на заседании Госкомиссии решила дело. Конечно, вопрос о том, каким путем решалось уравнение там не поднимался. Военные - люди серьезные, им не до таких мелочей. Им важно знать, что будет и как будет и кто будет отвечать, если все будет не так, а иначе.

Можно ли использовать положительные результаты кабинетной игры в обоснование сложных расчетов для ответственной техники? Во всяком случае, акты и протоколы были подписаны и ответственность легла на плечи отважных начальников - их убедила именно игра!

Через четыре месяца после этого станция полностью выполнила поставленную перед ней задачу, а поведение всех систем по данным телеметрической информации совпало с расчетным с точностью до полутора процентов, так были спасены миллиарды государственных рублей, затраченных на программу полета. Восторги молодого специалиста по этому поводу можно не описывать, а отважные начальники лукаво усмехались, принимая поздравления на самом высоком уровне.

История получила, кстати, неожиданное продолжение. Мы ведь сказали, что в уравнение входило много неизвестных параметров, для определения которых требовались длительные и дорогие испытания техники. Но раз уж приближенное решение было найдено, то интересно стало поглядеть и на то, чему эти параметры оказались равны. Так вот, два из них оказались настолько далеки от того, что предсказывала теория, что это вызвало массу сомнений. Раз за разом случайным поиском искали новые решения, но эти два параметра все равно отличались от теории раза в два. Впоследствии, когда экспедиция закончилась успешно и подтвердила расчет, стали проверять теорию и выяснили, что в ее основу была положена модель начала 60-х годов, которая успела устареть. Скорректировав модель и построив новую теорию, с удивлением обнаружили, что эти параметры оказались почти такими, как показало решение уравнения со многими неизвестными.

2.

Год 1990-ый. Только что появился на свет "ИНФОРКОМ" и сразу предложил своим клиентам первый десяток методических разработок.

Встал щепетильный вопрос определения цены на каждую. Большую цену назначить нельзя. Деньги, затраченные на их подготовку и печать не вернуться и фирму ждет мгновенная смерть. Малую цену назначить нельзя, ибо если спрос превысит очень ограниченные физические возможности коллектива, то не хватит подготовленного тиража, не хватит сил и времени на транспортировку и отправку, не хватит места для хранения, сорвется подготовка новых работ. Пойдут срывы заказов, потеря доверия клиентов и в итоге тоже смерть.

Два десятка неизвестных (цена на каждую из разработок и их тиражи) вошли в

несколько уравнений, увязывающих физические возможности коллектива, полиграфические мощности дружественного предприятия, почтовые возможности дружественного отделения связи и потенциальные финансовые возможности неизвестных будущих клиентов.

Расчет проводился на этот раз уже не на ЕС-1022, а на "Спектруме", что намного удобнее, по той же известной теперь Вам методике и в результате его были получены цены, которые надо назначить на ту или иную разработку и примерный ожидаемый тираж (для каждой - свой). По прошествии года исключительно ради интереса было проведено сравнение того, что дал компьютер с тем, что оказалось на самом деле. Самая грубая ошибка составила всего 35 процентов. Так еще раз удачно сработала эта увлекательная, но очень полезная игра.

В заключение хочется еще раз напомнить тем, кто нас читает - игра дело серьезное! Играйте на здоровье, но старайтесь при этом одновременно и обогащаться знаниями - кто знает, когда это пригодится!

SINCLAIR LOGO

(Продолжение)

Начало см. на с. 69-74, 90-95, 178 - 182.

7. И снова об обработке списков.

Мы уже видели с Вами некоторые процедуры, выдающие какой-то результат и назвали их операциями. С помощью команды OUTPUT Вы можете сами создавать такие операции. В этом случае команда OUTPUT определяет то значение, которое должно быть возвращено в качестве результата. Например, если мы хотим создать процедуру, которой нам не хватает в стандартной версии ЛОГО, такую, которая будет вычислять разность двух чисел аналогично тому, как процедура SUM вычисляет их сумму, то мы делаем так:

```
TO DIFFERENCE: NUM1: NUM2
  OUTPUT :NUM1 - :NUM2
END
```

и тогда:

```
PRINT DIFFERENCE 7 5
даст результат:
2
```

Когда ЛОГО встречает команду OUTPUT, он передает то значение, которое находится справа от слова OUTPUT и останавливает процедуру. Поэтому DIFFERENCE 7 5 имеет значение 7-5.

Вы можете в одной процедуре иметь несколько команд OUTPUT, но в этом случае они должны стоять в разных ветвях IF, поскольку первая же встреченная команда OUTPUT прервет исполнение процедуры, действительно, нам ведь нужен только один результат ее работы.

Например, Вы можете сделать так, чтобы процедура вычисления разности двух чисел всегда выдавала положительный результат. Для этого надо, чтобы оба числа сравнивались между собой и из большего вычиталось меньшее.

```
TO DIFFERENCE :NUM1 :NUM2
  IF :NUM1> :NUM2 [OUTPUT :NUM1-:NUM2]
  OUTPUT :NUM2 - :NUM1
END
PRINT DIFFERENCE 10 3
7
PRINT DIFFERENCE 2 6
4
```

Можно, конечно, было записать и так:

```
IF :NUM1> :NUM2 [OUTPUT :NUM1-:NUM2][OUTPUT :NUM2-:NUM1]
```

но в этом нет особой необходимости, если исполняется первая команда OUTPUT, то и процедура закончится, а до второй команды OUTPUT дело и не дойдет.

Точно так же можно сконструировать и процедуру, выдающую по команде OUTPUT слово или список. Например, мы хотим узнать, начинается ли слово с гласной буквы или нет. Если да, то в качестве артикля перед ним надо ставить по правилам английской грамматики не "A", а "AN".

Поскольку нас интересует только первая буква в слове, то неплохо проверить, не является ли слово пустым списком.

Вот пример, который выдает значение "AN", если слово начинается с гласной, или выдает "A", если слово начинается с иной буквы или выдает пустой символ, если слово -

пустое.

```
TO ARTICLE :AWORD
  IF EMPTY? :AWORD [OUTPUT ""]
  IF MEMBER? FIRST :AWORD[A E I O U][OUTPUT "AN"]
  OUTPUT "A"
END
```

Попробуйте сами изменить процедуру ARTICLE так, чтобы она могла работать не только с прописными, но и со строчными буквами тоже. Бывают случаи, когда операциями надо пользоваться очень осторожно. Прежде всего, некоторые операции могут иметь побочные эффекты, т.е. они могут вести себя как команды. Например, мы можем создать операцию, которая запросит имя пользователя и выдаст первое из введенных в нее слов в качестве результата.

```
TO GETNAME
  PRINT [WHAT IS YOUR NAME]
  OUTPUT FIRST READLIST
END
```

Поэтому:

```
MAKE "YOU GETNAME
```

даст сообщение

```
WHAT IS YOUR NAME
```

и будет ждать ввода с клавиатуры, прежде чем передаст первое прочитанное слово в качестве результата операции. Все это выглядит достаточно просто, но что будет, если Вы попробуете:

```
PRINT GETNAME
```

Обратите внимание на то, что внутри процедуры GETNAME тоже имеется команда печати PRINT. ЛОГО вычисляет результат работы процедуры справа налево и потому сначала исполнит GETNAME, а затем передаст результат команде PRINT.

То же самое произойдет и в команде:

```
PRINT SENTENCE "HELLO GETNAME
```

Здесь и GETNAME и "HELLO обеспечивают входные данные для команды SENTENCE, которая передает свой результат команде PRINT.

Попробуйте:

```
PRINT (SENTENCE GETNAME "ALIAS GETNAME)
```

и посмотрите, получится ли в итоге именно тот результат, которого Вы ждете.

Второй момент, о котором нужно всегда помнить, заключается в том, что всякий раз, когда вызывается исполнение операции, она работает, как в первый раз.

```
PRINT DIFFERENCE 7 5
2
PRINT DIFFERENCE 7 5
2
```

Так все и должно быть и ничего иного мы и не ждем (но при многократных повторных вычислениях на это уходит много времени). Сколько бы раз Вы не вызывали GETNAME, эта процедура всегда будет делать запрос и ждать ввода с клавиатуры. Если же Вам надо запомнить то имя, которое было введено в самый первый раз, то Вам надо присвоить его символьной переменной и обращаться к ней всякий раз, когда Вам будет необходимо:

```
MAKE "YOU GETNAME
PRINT SENTENCE "HELLO :YOU
PRINT SENTENCE [I AM GLAD TO MEET YOU]: YOU
```

Уровни процедур.

Когда одна процедура вызывает другую, мы говорим, что мы перешли на второй уровень процедуры. Если и эта процедура, в свою очередь, вызывает еще одну, то это уже третий уровень и т. д. Вот, например, три процедуры:

```
TO PROC1
  PRINT [LEVEL 1]
  PROC2
END
TO PROC2
  PRINT [LEVEL 2]
  PROC3
END
TO PROC3
  PRINT [LEVEL 3]
END
```

Здесь PROC1 - это первый уровень. Когда она вызывает PROC2, мы переходим на второй уровень. Когда же она вызывает PROC3, то это уже третий уровень, после чего Вам надо выполнить три команды END, STOP или OUTPUT, прежде чем мы вновь вернемся на верхний уровень (нулевой уровень).

Нулевой уровень в ЛОГО - это командный уровень, именно на нем Вы находитесь, когда перед Вами на экране изображено приглашение "?". Есть и специальная команда TOPLEVEL, которая позволяет сразу вернуться на верхний уровень, независимо от того, где Вы находитесь. Впрочем, этой командой пользоваться не рекомендуется, разве что в случае крайней необходимости.

Уровень, на котором Вы находитесь, никак не зависит от содержания тех процедур, которые Вы исполняете, а зависит только от того, сколько процедур, вложенных друг в друга, Вы исполнили. Представьте себе, что каждая встреченная на Вашем пути процедура - это ступенька вниз, а каждая встреченная команда END, STOP или OUTPUT - это ступенька вверх. Поэтому, если находясь на самом верхнем (командной) уровне Вы вызовете PROC3, то перейдете на первый уровень, хотя по своему содержанию PROC3 устроена так, что выдаст сообщение <LEVEL3>.

Интересны рекуррентные процедуры. Они могут уходить в глубину на очень много уровней.

```
TO DOLEVEL :N
  PRINT SENTENCE "LEVEL :N
  IF :N<10[DOLEVEL :N+1]
END
```

Тогда команда DOLEVEL 1 даст следующую печать:

```
LEVEL1
LEVEL2
LEVEL3
LEVEL4
LEVEL5
LEVEL6
LSVEL7
LEVEL8
LEVEL9
```

Когда же, в конце концов, N станет равно 10, мы выполним команду END на 9-ом уровне, потом на 8-ом и так далее до верхнего уровня, в этом можно и убедиться самому, если изменить DOLEVEL так:

```
TO DOLEVEL :N
  PRINT SENTENCE -LEVEL :N
  IF :N<10 [DOLEVEL:N+1]
  PRINT "FINISH
END
```

```

DOLEVEL 1
  LEVEL1
  LEVEL2
  LEVEL3
  LEVEL4
  LEVEL5
  LEVEL6
  LEVEL7
  LEVEL8
  LEVEL9
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH
  FINISH

```

Глобальные и локальные переменные.

Когда слово используется в качестве входного параметра в процедуру (при определении процедуры), тогда при вызове процедуры это слово должно принимать то значение, которое было задано при вводе. Простой пример:

```

TO SAYNUM :N
  PRINT SENTENCE [N IS] :N
END

```

И тогда:

```

SAYNUM 5
  дает ожидаемый результат:
  N IS 5

```

Теперь попробуйте так:

```

MAKE "N 7
SAYNUM 5
  N IS 5

```

А теперь попробуйте:

```

PRINT N

```

На этот раз результат будет неожиданным - 7. ЛОГО запомнит то значение, которое имела переменная N до вызова процедуры и отложит его до тех пор, пока процедура не закончится.

Итак, мы можем считать, что переменная N в процедуре SAYNUM является локальной для данной процедуры, т.е. SAYNUM имеет свое собственное значение N, которое может быть использовано только внутри этой процедуры. Те же слова, которые не указаны при определении процедуры в качестве ее параметров, являются глобальными (т.е. значения этих переменных одинаковы во всех частях программы). Отсюда вытекает одно важное следствие для того случая, когда используем несколько уровней вложения процедур. Мы покажем его на примере трех процедур PROC1, PROC2, PROC3.

```

TO PROC1 :N
  PRINT SENTENCE "PROC1 :NUM
  PROC2 5
  PRINT SENTENCE [PROC1 AGAIN] :NUM
END
TO PROC2 :N
  PRINT SENTENCE "PROC2 :NUM
  PROC3 7
  PRINT SENTENCE [PROC2 AGAIN] :NUM
END
TO PROC3 :N

```

```
PRINT SENTENCE "PROC3 :NUM
END
```

Здесь процедуры 1 и 2 печатают значение своего параметра прежде, чем вызывать следующую процедуру, поэтому:

```
PROC1 3
PROC2 5
PROC3 7
PROC2 AGAIN 5
PROC1 AGAIN 3
```

Итак, ЛОГО "помнит", какому уровню процедур, какое значение NUM соответствует. Проще всего, по-видимому, представлять себе локальные переменные, как совершенно новые переменные, когда Вы находитесь внутри процедуры, несмотря на то, что они могут иметь то же имя, что и переменные, встречающиеся вне этой процедуры.

Если локальная переменная изменяется внутри процедуры, то это не имеет никаких эффектов на все одинаковые переменные вне этой процедуры. Если же изменяется глобальная переменная, то это изменение действует всюду, например:

```
TO TEST :NUM
MAKE "NUM :NUM+1
PRINT :NUM
MAKE "OTHER :OTHER+1
END
MAKE "NUM 5
MAKE "OTHER 7
TEST 5
6
PRINT :NUM
5
PRINT :OTHER
8
```

Это имеет важное значение для рекурсии, Вы можете положиться на ЛОГО, когда он закончит рекуррентную процедуру в том смысле, что ЛОГО "помнит", где он был и каковы были там значения локальных переменных.

Команда OUTPUT для рекуррентных процедур.

Вы можете использовать процедуры рекуррентно, поставив в команду OUTPUT имя самой этой же процедуры, но если Вам надо, чтобы эта процедура все-таки закончилась и при этом выдала бы какой-то результат. Вам надо иметь еще хотя бы один OUTPUT, в котором не происходит рекуррентного вызова этой процедуры еще раз.

Чтобы все это стало ясным, мы изменим ранее использованную процедуру GETNAME так, чтобы она запрашивала имя пользователя и выдавала бы первое слово введенного имени, если это возможно, а если нет, то печатала бы соответствующее сообщение и вызывала бы саму себя для повторения попытки. Правда, имейте в виду, что это приведет и к повторной печати запроса. Мы не будем здесь пользоваться конструкцией FIRST READLIST.

Поскольку оператор FIRST не любит пустые списки (и это естественно) и потому сначала проверим, не является ли введенный список пустым, а только потом будем к нему применять оператор FIRST.

```
TO GETNAME
PRINT [ПОЖАЛУЙСТА НАЗОВИТЕ ВАШЕ ИМЯ]
MAKE "YOU READLIST
IF EMPTYR :YOU[PRINT[ХОТЯ БЫ ОДНО ИМЯ У ВАС ДОЛЖНО БЫТЬ]]
MAKE "YOU FIRST :YOU
IF NUMBERP:YOU[НОМЕР ВМЕСТО ИМЕНИ БЫВАЕТ ТОЛЬКО У ЗАКЛЮЧЕННЫХ] OUTPUT GETNAME]
OUTPUT :YOU
END
```

Это же можно проиллюстрировать на примере процедуры, входными параметрами которой будут число и слово, а в качестве результата будет слово, в которое будут вводить

столько символов, сколько указано входным числом. Итак:

```
START 5 "RECURSIVE
```

выдаст

```
RECUR.
```

Если :N - это число, а :INWORD - входное слово, то мы хотим получить от него :N первых символов. Теперь мы можем выделить из слова первый символ. После этого можно рекуррентно вызывать процедуру START до тех пор, пока еще не будут выданы :N-1 символов от оставшейся части входного слова :INWORD, т.е.

```
START :N-1 BUTFIRST :INWORD,
```

а затем поместить первый символ в новое слово:

```
WORD FIRST :INWORD
```

```
START :N-1 BUTFIRST :INWORD,
```

после чего выдать его в качестве результата с помощью OUTPUT.

Пока все идет хорошо, но встает традиционный вопрос: "Где же остановить рекурсию?" Поскольку N постепенно уменьшается и указывает на то, сколько еще символов мы хотим получить, то остановиться надо тогда, когда :N равно нулю и выдать "пустое слово" для обеспечения перехода START вверх на предыдущий уровень.

```
TO START :N :INWORD
```

```
IF :N= [0 OUTPUT"]
```

```
OUTPUT WORD FIRST :INWORD
```

```
START :N-1 BUTFIRST :INWORD
```

```
END
```

```
PRINT START 5 "RECURSIVE
```

```
RECUR
```

Давайте проследим ход работы процедуры с помощью следующей таблицы (табл. 1):

Таблица 1

Уровень	:N	: INWORD	FIRST : INWORD	BUTFIRST : INWORD	OUTPUT
1	5	RECURSIVE	R	ECURSIVE	
2	4	ECURSIVE	E	CURSIVE	
3	3	CURSIVE	C	URSIVE	
4	2	URSIVE	U	RSIVE	
5	1	RSIVE	R	SIVE	
6	0				"
5	1	RSIVE	R		R=WORD"R"
4	2	URSIVE	E		UR=WORD"U"R
3	3	CURSIVE	C		CUR=WORD"C"UR
2	4	ECURSIVE	U		ECUR=WORD"E"CUR
1	5	RECURSIVE	R		RECUR=WORD"R"ECUR

Вложенные списки.

В качестве элементов списков в ЛОГО тоже могут быть списки. В этом нет ничего удивительного. Если мы составим список того, что лежит у нас в кармане, то можем получить:

```
[МОНЕТА КАРАНДАШ ЛАСТИК СПИСОК ПОКУПОК]
```

Обратите внимание на то, что объекты, входящие в список покупок вовсе не являются элементами списка того, что лежит у нас в кармане, таким образом, список содержимого кармана имеет 4 элемента, несмотря на то, что список покупок сам является списком:

```
[ХЛЕБ МОЛОКО МЯСО МАРКИ]
```

а ХЛЕБ не является вещью, находящейся в кармане.

Более того, ЛОГО даже допускает создать такой список содержимого кармана:

```
[МОНЕТА КАРАНДАШ ЛАСТИК [ХЛЕБ МОЛОКО МАСЛО МАРКИ]]
```

и в этом списке по-прежнему всего 4 элемента, несмотря на то, что четвертый элемент представляет из себя список из 4-х элементов.

Сколько же элементов в следующем списке?

```
[ [ХЛЕБ МОЛОКО МАСЛО МАРКИ] [ДЖЕЙН ДЭВИД РОЗМАРИ] [СТИРКА ГЛАЖКА]
[8_МАЯ 20_ИЮНЯ 1_ИЮЛЯ 10_СЕНТЯБРЯ 23_НОЯБРЯ] ]
```

В этом списке 4 элемента. Все они - тоже списки. Проще было бы дать им имена:

```
[ПОКУПКИ ДРУЗЬЯ ЗАБОТЫ ДАТЫ]
```

Вы составили такой список для себя, теперь Вы можете составить аналогичный список для друга и еще раз для кого-то и тогда получите список списков:

```
[МОЙ ТВОЙ ЧУЖОЙ]
```

Здесь МОЙ - это [ПОКУПКИ ДРУЗЬЯ ЗАБОТЫ ДАТЫ], точно так же ТВОЙ и ЧУЖОЙ. Так можно продолжать и дальше, вкладывая список один в другой, но наверное стоит остановиться и предоставить такую работу ЛОГО.

Процедуры создания списков

Мы уже использовали команду SENTENCE для того, чтобы создать список из двух и более элементов, которые могут быть словами или списками. Эта команда объединяет их в один список, снимая внешние квадратные скобки и переписывая весь список заново. Например:

```
(SENTENCE [ВОТ ВАМ][ПРИМЕР ДЕЙСТВИЯ][ОПЕРАТОРА SENTENCE])
```

выдает в качестве результата:

```
[ВОТ ВАМ ПРИМЕР ДЕЙСТВИЯ ОПЕРАТОРА SENTENCE]
```

Сейчас же мы с Вами рассмотрим три новых оператора, по-разному создающих списки.

FPUT (это First PUT). Имеет два входных параметра. Второй параметр - список. Команда FPUT создает новый список, в котором первый элемент берется из первого входного параметра, а остальная часть списка из второго.

Например:

```
FPUT "РАЗ[ДВА ТРИ ЧЕТЫРЕ]
```

выдаст список:

```
[РАЗ ДВА ТРИ ЧЕТЫРЕ]
```

В отличие от SENTENCE здесь не происходит слияния внешних квадратных скобок при объединении списков, например:

```
FPUT [РАЗ ДВА][ТРИ ЧЕТЫРЕ]
```

дает:

```
[ [РАЗ ДВА] [ТРИ ЧЕТЫРЕ]
```

Можете считать, что FPUT берет свой первый параметр и ставит его в первую позицию второго параметра.

Другой оператор, который мы рассмотрим, - это LPUT (Last PUT). Он работает совершенно аналогично, но с одним исключением. Первый параметр LPUT ставится последним элементом в списке, выраженном вторым входным параметром.

```
LPUT [ПОСЛЕДНИЕ СЛОВА][ЭТО]
```

дает:

[ЭТО [ПОСЛЕДНИЕ СЛОВА]]

Третий оператор - LIST. Он имеет два или более входных параметров и выдает их в виде списка, независимо от того, являются ли они списками или словами.

```
LIST "БАНК [ХЛЕБ МОЛОКО МАСЛО МАРКИ]
```

имеет два входных параметра и выдает список следующего содержания:

```
[БАНК [ХЛЕБ МОЛОКО МАСЛО МАРКИ]
```

Если LIST имеет более двух входных параметров, то они должны быть заключены в круглые скобки.

```
(LIST "БАНК [ХЛЕБ МОЛОКО МАСЛО МАРКИ][ДЖЕИН ДЭВИД РОЗМАРИ]).
```

выдает список:

```
[БАНК [ХЛЕБ МОЛОКО МАСЛО МАРКИ] [ДЖЕИН ДЭВИД РОЗМАРИ]]
```

Можете полагать, что команда LIST просто берет все свои параметры и ставит вокруг них дополнительную пару квадратных скобок.

ВНИМАНИЕ! Когда LIST используется более, чем с двумя параметрами и один из них равен самому списку LIST, то в этих случаях оператор не всегда работает правильно. Он также может вызвать проблемы, если включен в квадратные скобки при операторе REPEAT или в список IF. Очень сложные конструкции, таким образом, следует создавать постепенно.

Сортировка чисел.

Теперь мы посмотрим, как можно использовать списки для создания сложных структур данных и для некоторых других практических целей. Сначала мы рассмотрим вопрос сортировки последовательности чисел, введенных с клавиатуры, сортировка данных это очень важный вопрос, в котором проявляется мощь компьютеров. Существует много методов для выполнения сортировки, но мы остановимся на одном из них.

Мы будем хранить числа в структуре, называемой "деревом". На рис. 1 показан пример такого дерева.

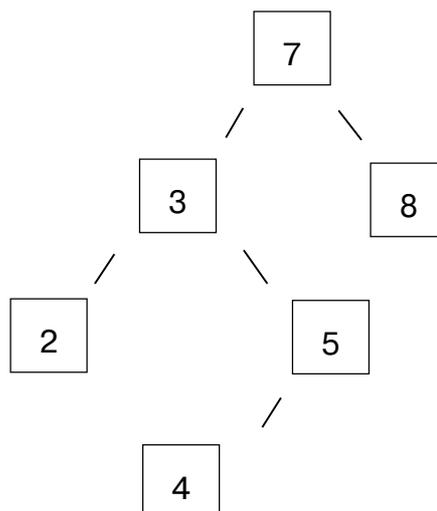


Рис. 1

Если не очевидно, почему эта структура получила такое название, переверните рисунок вверх ногами. Каждое обведенное число называется "узлом", в каждом из узлов дерево может ветвиться (максимум на две ветви).

Посмотрите, как это организовано. Возьмем, например, число 6 и введем его в наше дерево. Начнем с вершины и в каждом узле будем сравнивать наше число с тем, что

содержится в данном узле. Если наше число меньше, то из узла пойдём по левой ветви, а если больше - то по правой.

Представим, что число 6 помещается на вершину дерева, поскольку $6 < 7$, то идем вниз по левой ветви к числу 3. Здесь у нас $6 > 3$ и продолжаем путь по правой ветви к числу 5. Раз $6 > 5$, то и здесь надо пойти по правой ветви, но на сей раз у нас правой ветви нет и ее надо сделать. Просто пририсовываем правую ветвь к узлу 5 и поставим в ее конце узел 6.

Итак, каждый узел может иметь две ветви, одну влево для чисел, которые меньше, чем значение в узле и одну вправо для чисел, которые больше, чем значение в узле, простой вопрос "наше число больше, чем значение в узле или меньше?" определяет, по какой ветви нам идти.

Точно так же можно построить дерево и для списков. Каждый узел может быть представлен списком из трех элементов.

Средний элемент - это номер узла, а два других элемента - тоже списки. Первый элемент определяет левую ветвь дерева, а последний - правую, каждая ветвь выстраивается аналогично. Если из узла ветвь не исходит, то ей соответствует пустой список. Таким образом, пустые списки служат как бы маркерами, показывающими, что мы дошли до конца данной ветви.

Вы, наверное, не удивитесь, когда узнаете, что такие конечные узлы называют "листьями", а исходный узел - "корнем".

Дерево для структуры списков показано на рис. 2.

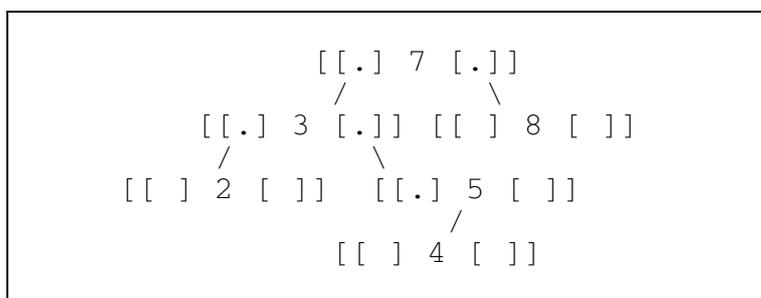


Рис.2

В тех узлах, где список не пустой, стоит "точка" [.] и стрелка, показывающая содержимое данного списка, поэтому несмотря на то, что списки находятся один внутри другого, такой "развернутый" вид действительно демонстрирует похожесть на настоящее дерево.

Давайте еще раз посмотрим, как число 6 вписывается в это дерево. Рассмотрим корневой узел на вершине диаграммы, он имеет 3 элемента, из которых средний - число 7. Новое число 6 меньше и потому мы идем по левой ветви, рассматривая левый список. Этот список имеет средний элемент, равный 3. Поскольку 6 больше, то мы ищем правое продолжение, новый список имеет средним элементом 5. Раз 6 больше, чем 5, мы ищем опять правое продолжение. На этот раз список пуст, мы дошли до конца ветви и теперь нашли место, в которое должно быть помещено число 6. Поскольку это последняя точка, то мы отмечаем ее пустыми списками: `[[] 6 []]`.

Итак, существуют только два правила для того, чтобы поместить новое число в дерево.

1) Если число помещается в узел, в центре которого уже есть число, то эти числа сравниваются и, если новое число меньше, то оно помещается в начальный список, а если больше, то в конечный.

2) Если число помещается в пустой список, оно становится в средний элемент, а справа и слева добавляются по пустому списку.

Если мы начнем с пустого списка и будем вводить числа в порядке 7,3,5,8,... то у нас будет создаваться список так, как показано в табл. 2. На каждом шагу мы показываем только новые вошедшие в список элементы, дабы сделать структуру проще и нагляднее.

```

ШАГ
1      [ ]
2      [ ]3[ ]7[ ]
3      [ ]3[ ]
4      [ ]5[ ]
5      [ ]8[ ]

```

Как видите, числа ложатся в список в правильном порядке.

Теперь рассмотрим, как сделать это на языке ЛОГО. Сначала нам нужна процедура, которая выдаст пользователю необходимые инструкции и создаст исходный пустой список, затем выполнит ввод чисел и распечатает упорядоченный список.

```

TO SORT
  PRINT [Введите числа]
  PRINT [Когда закончите, наберите END]
  MAKE "NUMLIST []
  INPUT
  PRINTINORDER :NUMLIST
END

```

Теперь рассмотрим процедуру INPUT. Она должна принять с клавиатуры первое слово и проверить, не число ли это. Мы используем тот факт, что это не число для того, чтобы сигнализировать о том, что ввод закончен. Если же это число, то его надо ввести в список и посмотреть, куда оно ложится, после чего сформировать новый список.

```

TO INPUT
  MAKE "NUM FIRST READLIST
  IF NOT NUMBERP :NUM [STOP]
  MAKE "NUMLIST RENEW :NUMLIST
END

```

Здесь мы впервые использовали операцию NOT. Она берёт то условие, которое стоит в качестве ее параметра и изменяет полученный результат с TRUE на FALSE или, соответственно, наоборот. Так что она работает достаточно просто и выдает результат TRUE (ИСТИНА), если :NUM числом не является.

Теперь нам нужно записать процедуру RENEW, которая проверит введенное число по всем имеющимся спискам и вставит его на свое законное место. Эта процедура должна создавать списки из трех элементов, обычно по результатам IF. Как было указано в предыдущей главе, LIST нельзя использовать с круглыми скобками внутри, так что мы начнем с того, что напишем процедуру, которая примет три параметра и сделает из них список:

```

TO LIST :A :B :C
  OUTPUT (LIST :A :B :C)
END

```

Теперь, когда мы проверяем введенное число по имеющемуся списку (назовем процедуру ALIST), у нас могут быть три возможности исхода:

- список может быть пустым, в этом случае мы заменяем его новым списком из трех элементов, внешние из которых являются пустыми списками, а средний - введенным числом;

```
LIST 3 []:NUM[]
```

- если вводимое число больше, чем средний элемент списка, то новый список замещает тот, который был. В нем первые два элемента остаются неизменными, а в качестве третьего вводится новый список, содержащий вводимое число:

```
LIST3 FIRST :ALIST ITEM2 :ALIST
```

```
RENEW LAST :ALIST
```

- если вводимое число меньше, чем средний элемент, то в новом списке меняется первый элемент:

```
LIST3 RENEW FIRST :ALIST ITEM2 :ALIST LAST :ALIST
```

Теперь мы пришли к процедуре RENEW:

```
TO RENEW :ALIST
  IF EMPTY :ALIST [OUTPUT LIST3[] :NUM[]]
  IF :NUM>ITEM2 :ALIST
    [OUTPUT LIST3 FIRST :ALIST
     ITEM2 :ALIST
     RENEW LAST .ALIST]
  [OUTPUT LIST3
   RENEW FIRST :ALIST
   ITEM2 :ALIST
   LAST :ALIST]
END
```

Как мы уже видели, в результате наши числа будут упорядочены по старшинству. Проблемой, однако, является большое количество квадратных скобок. К счастью, здесь помогает сила рекурсии. Для того, чтобы упорядочить список, состоящий из списка, числа и еще одного списка, нам надо распечатать первый список в упорядоченном виде, число и второй список в упорядоченном виде. Если список пуст, то вообще ничего печатать не надо:

```
TO PRINTINORDER :ALIST
  IF EMPTY :ALIST [STOP]
  PRINTINORDER FIRST :ALIST
  PRINT ITEM2 :ALIST
  PRINTINORDER LAST :ALIST
END
```

Когда Вы закончите ввод процедур, дайте команду SORT. Введите несколько чисел, по одному на строке. Заканчивайте ввод чисел словом END или любым другим удобным словом, но не числом.

После того, как числа будут распечатаны в отсортированном порядке. Вы можете попробовать:

```
PRINT :NUMLIST
```

Теперь Вы увидите, как список хранится в структуре "дерева". Если же Вы хотите увидеть, как создается новый отсортированный список. Вы можете попробовать вставить эту команду в процедуру INPUT.

Самообучающаяся игра.

Сейчас мы попробуем использовать структуру "дерева" для того, чтобы написать игру. Суть игры состоит в том, что один играющий задумывает некоторое животное, а другой должен его угадать. При этом он может задавать любые вопросы, но ответ может быть только "да" или "нет". Но самая интересная часть игры состоит в том, что программа способна самообучаться. В начале игры она знает только двух животных, а по мере того, как Вы с ней играете, она становится все более и более образованной.

Вся информация хранится в виде дерева. Каждый узел - это список, состоящий из трех элементов. Обычно средний элемент списка - вопрос, который надо задать игроку. Если ответ "да", то программа использует первый элемент списка (а он тоже список) для следующего хода. Если же ответ "нет", то программа использует последний элемент списка.

Мы неизбежно дойдем до последней точки. Она выявляется по тому факту, что и первый и последний элемент являются пустыми списками. В данном случае средний элемент - это уже не вопрос к играющему, а название животного.

Когда имя животного выдается игроку, то у него есть две возможности:

- если компьютер угадал верно, то программу не надо обучать;
- но если пользователь задумал другое животное, то программа запрашивает вопрос, с помощью которого можно различить то животное, которое задумал пользователь, и то, которое назвала программа. Это вопрос теперь становится средним элементом в новом узле, а названия животных - в узлах справа и слева. Опять же к каждому из этих названий добавляется справа и слева по пустому списку. Входная процедура должна ввести пользователя в суть игры и создать корневой список перед тем, как игра начнет свою работу.

```
TO ANIMAL
  PRINT " PRINT [Добро пожаловать на викторину]
  PRINT [Отвечайте только "ДА" или "НЕТ"]
  MAKE "NODE [[[]][УТКА][[]]
    [ОНО ПЛАВАЕТ?]
    [[[]][СВИНЬЯ][[]]]
  GO
END
```

Рабочая процедура GO просит играющего задумать животное, проверяет элементы своего списка и обеспечивает повтор всего процесса для другого животного.

```
TO GO
  PRINT [ЗАДУМАЙ ЖИВОТНОЕ]
  PRINT "
  MAKE "NODE TRY :NODE
  PRINT [ЕЩЕ РАЗ?]
  MAKE "YN FIRST READLIST
  IF YN= "YES [GO]
END
```

Процедура GO проверяет текущий узел. Это может привести к изменению списка, поэтому процедура TRY выдает новое содержание узла.

Необходимо предусмотреть две возможности. Средний элемент в узле может быть либо вопросом, который надо задать игроку, либо готовым ответом.

Программа различает эти два случая, проверив первый элемент. Если он пуст, то во втором элементе содержится ответ.

```
TO TRY :ANODE
  IF EMPTY FIRST :ANODE
    [OUTPUT GUESSANSWER]
    [OUTPUT PUTQUESTION]
  END
```

Теперь процедура GUESSANSWER должна угадать ответ и, если он верен, то изменений делать не надо, а если нет, то надо сконструировать новый узел?

```
TO GUESSANSWER
  PRINT "
  PRINT SENTENCE [ЭТО]
  ITEM2 :ANODE
  MAKE "YN FIRST READLIST
  IF :YN= "YES
    [PRINT [Я ТАК И ДУМАЛ]
    OUTPUT :ANODE]
    [OUTPUT NEWNODE]
  END
```

Процедура NEWNODE запрашивает правильный ответ и конструирует новый узел для двух животных, используя при этом полученный от пользователя вопрос. "Новое" животное должно быть помещено в новый список между двумя пустыми списками. "Старое" животное уже организовано таким образом в текущем узле.

Здесь нам вновь будет нужна процедура LIST3, о которой мы говорили выше:

```
TO NEWNODE
PRINT [Сдаюсь! что это за животное?]
MAKE "ANIMAL1 READLIST
MAKE "NEWPART[LIST[1:ANIMAL1[]]]
MAKE "ANIMAL2 ITEM2 :ANODE
PRINT [SENTENCE[Дайте мне вопрос, чтобы отличить] :ANIMAL1 "и :ANIMAL2]
MAKE "QUESTION READLIST
PRINT [SENTENCE[и для ] :ANIMAL1 [ответ будет да?]]
MAKE "YN FIRST READLIST IF :YN="ДА"[OUTPUT LIST3 :NEWPART :QUESTION :ANODE] [OUTPUT LIST3
:ANODE :QUESTION :NEWPART]
END
```

Это самая сложная процедура в программе, но ключ к ее пониманию лежит в двух операциях OUTPUT, записанных в ее конце.

```
TO PUTQUESTION
PRINT ""
PRINT ITEM2 :ANODE
MAKE "YN FIRST READLIST
IF :YN=" ДА [OUTPUT LIST3
TRY FIRST :ANODE
ITEM2 :ANODE
LAST :ANODE]
[OUTPUT LIST3 FISTST :ANODE
ITEM2 :ANODE
TRY LAST :ANODE]
END
```

И здесь тоже основная смысловая часть процедуры заключена в двух последних операциях OUTPUT.

Теперь, если все сделано правильно. Вы можете дать команду ANIMAL и, работая с программой, увидеть, как происходит ее самообучение.

8. Некоторые математические операции.

ЛОГО различает два различных вида чисел: целые (например 1, -23, 245 и т.д.) и десятичные (0.67, 28.91 и т.п.). Обычно Вам не надо думать о том, что между ними есть разница, вероятно, единственный случай, когда различие между ними существенно, это тогда, когда Вы пытаетесь ввести десятичное число в процедуру, которая ожидает в качестве своего параметра целое число. И даже в этом случае, как правило, ЛОГО предупредит Вас.

Операция INT (INTEGER) принимает в качестве входного любое число, а свой результат выдает в виде целого. Она, таким образом, отрезает десятичную часть числа.

```
PRINT INT 3,14
3
PRINT
6
PRINT INT -2.1
-1
```

Обратите внимание на то, что хотя 6,999 очень близко к числу 7, результатом все-таки будет число 6. Этот процесс всегда ведет к округлению вниз.

В некоторых случаях такая особенность может быть полезной. Например, иногда бывает важным найти результат деления одного числа на другое и образующийся при этом остаток.

Может это быть полезным, когда Вы проверяете, является ли данное число целым или нет. Если оно целое, то применение к нему функции INT уже не изменит его.

```
TO TESTINT1 :NUM
IF :NUM-INT :NUM
```

```
[PRINT "YES] [PRINT "NO]
END
```

Таким же способом можно определить и делится ли одно число на другое нацело.

```
TO TESTDIV :NUM1 :NUM2
  IF (NUM1/NUM2) = INT(:NUM1/:NUM2)
    [PRINT "YES] [PRINT "NO]
END
```

Если же Вам на самом деле нужно ближайшее целое число, то Вы можете воспользоваться операцией ROUND. Она тоже может принимать любое число в качестве входного и выдает ближайшее целое число.

```
PRINT ROUND 3.14
  3
PRINT ROUND 6.99
  7
PRINT ROUND -2.9
 -3
```

Эта операция берет свое название из термина rounding ("округление"). Она может быть использована для того, чтобы выдавать ответ с заданной степенью точности, с небольшими усилиями ее можно использовать и для получения результата с заданным количеством десятичных знаков.

Так, например, если Вам надо округлить 3.14159 до двух десятичных знаков, то Вы можете умножить это число на 100 (314.159), округлить его (314) и снова разделить на 100 (3.14). Эту операцию можно конвертировать в рекуррентную процедуру, на каждом шагу умножая исходное число на 10. Если количество десятичных чисел равно нулю, то число просто округляется.

```
TO CORRECT :NUM :DECPLS
  IF DECPLS=0 [OUTPUT ROUND :NUM]
  OUTPUT (CORRECT :NUM*10 :DECPLS -1)/10
END
```

Есть еще одна операция, которая всегда выдает в результате своей работа целое число. Это операция REMAINDER. Она требует два входных параметра и выдает остаток от деления нацело первого числа на второе.

```
PRINT REMAINDER 12 5
  2
```

Эта операция обеспечивает еще один путь для проверки делится ли одно число на другое нацело. Если это так, то результат операции равен нулю.

ИСТИНА и ЛОЖЬ (TRUE и FALSE)

Мы можем писать процедуры, которые проверяют условия и выдают в качестве результата TRUE или FALSE, это можно сделать одним из двух возможных способов, Вы можете выдавать результат в виде слова TRUE, FALSE и т.п., например:

```
TO DIVISIBLE :A :B
  IF 0=REMAINDER :A :B
    [OUTPUT "TRUE] [OUTPUT "FALSE]
END
```

Второй способ - получение ответа сразу в результате проверки и немедленной выдачи результата:

```
TO DIVISIBLE :A :B
  OUTPUT 0=REMAINDER :A :B
END
```

Здесь, если действительно остаток от деления :A на :B равен нулю, результат равен TRUE, а если нет - то результат равен FALSE.

Такой результат можно напрямую использовать для оператора IF.

```
IF DIVISIBLE 20 5 [PRINT "YES] [PRINT "NO]
```

Фактически Вы только что определили предикативную процедуру и чтобы это подчеркнуть, ее можно назвать DIVISIBLEP.

Иногда мы сталкиваемся с тем, что будет ли результат операции иметь значение TRUE или FALSE, зависит от одного или большего количества условий, ЛОГО обеспечивает три операции, работающих с предикатами (условиями), мы уже рассмотрели NOT, изменяющую результат условия на противоположный. Есть еще операция AND, которая дает в результате значение TRUE, если все входящие выражения имеют значение TRUE. Обычно операция AND обрабатывает два условия, но их может быть и больше, если применить круглые скобки.

```
PRINT AND 1<2 3<7
```

```
TRUE
```

Другая операция - OR (ИЛИ). Она дает в результате TRUE, если хотя бы одно из входящих выражений имеют значение TRUE. Так же, как и AND, операция OR может иметь более двух входных выражений.

При аккуратной работе можно объединять AND и OR в одном выражении. Например, Вы можете прийти ко мне на день рождения, если Вас зовут Фред или Вы женщина и к тому же красивая.

Возведение в степень и извлечение корня.

Узнать квадрат числа можно умножив его само на себя, другие степени числа легко находятся с помощью рекурсии.

```
TO POWER :A :B
  IF :B<2 [OUTPUT :A]
  OUTPUT (POWER :A :B -1)*A
END
```

Для вычисления корней квадратных из числа, ЛОГО имеет специальную функцию SQRT.

```
PRINT SQRT 25
5
```

Мы можем воспользоваться теоремой Пифагора для вычисления расстояния между двумя точками с известными координатами (координата X и координата Y). В учебниках по математике Вы найдете, что расстояние между двумя точками с координатами [X1, Y1] и [X2, Y2] можно найти, как

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Рис. 3

```
TO DISTANCE :POS1 :POS2
  MAKE "XDIF FIRST :POS1-FIRST :POS2
  MAKE "YDIF LAST :POS1-LAST :POS2
  OUTPUT SQRT(XDIF*XDIF+YDIF*YDIF)
END
```

Поиск простых чисел.

Простое число - это число, которое нацело делится только на единицу или само на себя. Проблема поиска простых чисел волнует математиков уже несколько тысяч лет. Пока не найдена формула, по которой можно было бы найти все простые числа, а те формулы, которые уже известны, выдают определенную последовательность из простых чисел, но начиная с какого-то числа выдают и непростые числа (См. ZX-РЕВЮ-91, №7,8. с. 133).

Хоть и не существует вполне надежной формулы для простых чисел, зато существует метод. Мы можем просто брать числа по очереди и проверять, делятся ли они нацело на другие числа, которые меньше, чем оно само. Задача упрощается тем, что нужно проверять не все числа, а только до корня квадратного из исходного числа. Ведь если оно делится на что-то, что больше, чем квадратный корень, то в результате получим частное, которое меньше, чем квадратный корень, а эти числа мы уже проверили.

```
TO TESTPRIME :NUM :N
  IF :N>SQRT :NUM[OUTPUT "TRUE]
  IF DIVISIBLE :NUM :N[OUTPUT "FALSE]
  OUTPUT TESTPRIME :NUM :N+1
END
```

Проверку надо начинать с числа 2 - это наименьшее простое число.

```
TO PRIMIFORM :NUM
  IF TESTPRIME :NUM2[PRINT :NUM]
  PRIMIFORM :NUM+1
END
```

Теперь команда PRIMIFORM 2 будет распечатывать простые числа, начиная с числа 2 и выше.

Тригонометрические операции.

Если Вы уже знакомы с тригонометрическими функциями, то можете пропустить один абзац. Если же нет, то читайте дальше.

Если наша "черепашка" имеет значение HEADING, равное углу :ANGLE, то на каждом шаге она проходит разные расстояния вдоль экрана и поперек. Если "черепашка" пройдет в направлении своего движения единицу пути, то ее перемещение по горизонтали равно синусу угла :ANGLE, а перемещение по вертикали - равно косинусу угла :ANGLE. Если путь, пройденный "черепашкой" равен единице, то очевидно, что путь, пройденный ею по горизонтали, меньше единицы, то же и путь, пройденный по вертикали.

Может быть, Вас интересует не один шаг, пройденный "черепашкой", а сразу 10. В этом случае ее перемещение по горизонтали:

```
10*SINE :ANGLE
  а путь по вертикали:
10*COSINE :ANGLE
```

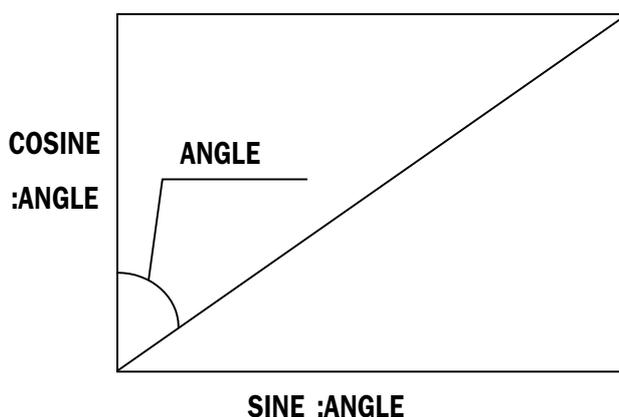


Рис. 4 Синус и косинус угла.

Математики пользуются еще двумя функциями: тангенс (это частное от деления синуса угла на его косинус) и котангенс (частное от деления косинуса на синус).

ЛОГО предоставляет возможность использования четырех операций: SINE, COSINE, TANGENT, COTANGENT, в сокращенной форме SIN, COS, TAN, COT. У ЛОГО есть большое преимущество перед другими языками программирования в том смысле, что ЛОГО принимает углы для этих операции в градусной мере, в то время как другие языки

программирования предпочитают иметь дело с радианами. Поэтому, если Вы привыкли к другим языкам и к радианной мере, то теперь вздохнете с облегчением. Если же Вы к ней не привыкли и никогда о ней не слышали, то тем более вздохнете с облегчением, поскольку можно о ней и не думать.

Кроме 4-х описанных выше операций, есть еще четыре: ARCSIN, ARCCOS, ARCTAN, ARCCOT, которые имеют прямо противоположное действие.

В качестве входного параметра они принимают число, а на выходе выдают угол в градусах, который имеет соответствующий синус (косинус, тангенс, котангенс).

Так, PRINT ARCCOS 0.5 даст Вам угол, косинус которого равен 0.5, т.е. 60 градусов.

Эти операции имеют огромное значение в изучении электроники, волновых явлений и вибраций. С помощью этих функций можно также исполнять очень красивые и интересные кривые. Рассмотрим, например, пару следующих процедур, которые изображают график функции косинус для углов от 0 до 360 градусов. Координаты X и Y здесь распечатываются через угол и его косинус, соответственно. Масштаб избран таким, чтобы изображение хорошо вписывалось в экран. Когда "черепашка" дойдет до границы экрана, Вы получите сообщение "turtle out of bounds".

```
TO SHOWCOS
  PENUP
  SETPOS [120 0] PENDOWN
  SETPOS [120 0] DO COS 0
END
```

```
TO DOCOS :ANGLE
  MAKE "X 2* :ANGLE/3-120
  MAKE "Y 80*COS :ANGLE
  SETPOS SE :X :Y
  DOCOS :ANGLE+15
END
```

Вы увидите колебания в вертикальной плоскости. Что произойдет, если мы добавим сюда еще и колебания по горизонтали? Вы получите фигуры, названные именем их первооткрывателя - фигуры Лиссажу.

```
TO LISSAJOU :A :B :C :D :INC
  SHOWTURTLE
  CLEARSCREEN
  PENUP
  LISS 0
END
```

```
TO LISS :ANGLE
  MAKE "X 120*COS(:A*:ANGLE+:B)
  MAKE "Y 80*COS(:C*:ANGLE+:D)
  SETPOS SE :X :Y
  PENDOWN
  LISS :ANGLE+:INC
END
```

Перо опустится после первой команды SETPOS (и после этого останется опущенным), поэтому "черепашка" пойдет в стартовую позицию без изображения линии. Чем выше значение INC, тем более угловатой будет полученная графика.

Попробуйте:

```
LISSAJOU 11 20 21 40 1
```

Введение в графику цвета позволяет получать очень интересные эффекты. Вот еще несколько примеров "черепашьей" графики:

```
TO CYCLO :K :ANGLE :SIZE
  FORWARD :SIZE*SIN(:K*HEADING)
```

```
RIGHT :ANGLE
CYCLO :K :ANGLE :SIZE
END
```

Попробуйте

```
CYCLO 1 10 10
CYCLO 4 5 20
```

И еще одна процедура:

```
TO LOBE :A :B :INC :ANGLE
  FORWARD 1
  RIGHT :A :B :INC :ANGLE+:INC
END
```

Попробуйте:

```
LOBE 2 3 9 0
LOBE 10 8 6 0
LOBE 2 8 6 0
LOBE 2 9 3 0
```

(Окончание в следующем номере).

ПРИМЕНЕНИЕ АСSEMBЛЕРА ДЛЯ СОЗДАНИЯ БЫСТРОРАБОТАЮЩИХ ПРОГРАММ

Перевод с английского Пашорина В.И.

Продолжение. Начало см. с. 9-12, 48-61, 97-103, 183-188

8. Команды ATTRIBUTE, SCREEN\$ и POINT

Если Вы программировали игры на БЕЙСИКе, то вероятно сталкивались с тем, что Вам необходимо знать в определенной игровой ситуации текущее расположение заданного символа на экране. Это, например, нужно при проверке условия - достигла ли ракета цели, либо при выполнении посадки на Луну. В БЕЙСИКе для этого существуют три способа:

- использовать оператор ATTRIBUTE, который проверяет значение INK, PAPER, FLASH и BRIGHT в заданном знакоместе экрана и выдает число, соответствующее этим параметрам;

- использовать оператор SCREEN\$, который определяет код символа в заданном знакоместе и выдает код этого символа, если он входит в диапазон 32...127. В противном случае считается, что ячейка пуста;

Примечание: процедуру в машинных кодах, которая может проверить не только символы ASCII, но и символы UDG-графики и символы блочной графики, т.е. из диапазона 32...164 мы привели в первом томе, посвященном графике для "Спектрума" - "Элементарная графика" на с. 99 под названием SCANNER. - ИНФОРКОМ.

- использование оператора POINT, который проверяет состояние одного пиксела экрана и выдает 1, если пиксел включен и 0, если он выключен.

Все эти три способа могут быть реализованы и в программах в машинных кодах с использованием специальных процедур ПЗУ.

8.1. ATTRIBUTE (строка, столбец).

Процедура ATTR, аналогичная по действию оператору ATTRIBUTE, находится в ПЗУ по адресу 9603. Перед ее вызовом необходимо записать координаты исследуемого знакоместа в регистровой паре BC (номер строки - в регистре B, а номер столбца - в регистре C). Результат проверки в виде числа 0...255 записывается на вершину стека калькулятора.

Определить атрибуты знакоместа, используя это число, можно по формуле:

ЧИСЛО=128*FLASH + 64*BRIGHT + 8*PAPER + INK значение

Это число можно переписать со стека калькулятора в регистр A, используя процедуру ПЗУ FP_TO_AA, находящуюся по адресу 11733, а затем сравнить его с заданным числом N с помощью команды CP N.

Пример реализации этого способа показан в программе 8.1. Программа проверяет каждое знакоместо экрана до тех пор, пока не найдет ячейку, атрибуты которой соответствуют PAPER 1, INK 7, FLASH 1, т.е. числу 143. После запуска этой программы на экране появляется ракета, медленно перемещающаяся по направлению к человечку. При совмещении ракеты с этим человечком происходит возврат в БЕЙСИК. Вы можете вместо команды возврата RET подключить процедуру, создающую звуковой эффект, сопровождающий попадание ракеты в человечка.

Такой способ проверки атрибутов символьных ячеек безусловно хорош, но требует, чтобы, например, все поражаемые объекты имели одинаковые атрибуты.

Листинг 8. 1.

АДРЕС	МЕТКА	МАШ. КОД	АСSEMBЛЕР	КОММЕНТАРИЙ
23760		ED 5B 7B 5C	ORG 23760 LD DE, (23675)	;Адрес первого символа

23764		21 42 5D	LD HL, DATA 3	; графики пользователя.
23767		01 08 00	LD BC, 8	; Конструкция символа UDG.
23770		ED B0	LDIR	; 8 байтов в символе.
23772		3E 02	LD A, 2	; Установка символа UDG-"A"
23774		CD 01 16	CALL 5633	; Канал экрана - 2.
				; Открываем канал печати на
				; экран.
23777		CD 6B 0D	CALL 3435	; Очистка экрана.
23780		3E 02	LD A, 2	; См. выше.
23782		CD 01 16	CALL 5633	
23785		11 2E 5D	LD DE, DATA1	; Данные для печати - DATA1.
23788		01 10 00	LD BC, 16	; Символов - 16.
23791		CD 3C 20	CALL 8252	; Вызов подпрограммы ПЗУ для
				; печати символьной строки.
23794		FD 36 57 03	LD (IY+87), 3	; (IY+87) указывает на
				; системную переменную PFLAG
				; (23697). Установка в ней
				; числа 3 означает включение
				; режима INVERSE.
23798		AF	XOR A	; Обнуление аккумулятора.
23799		32 40 5D	LD (COL), A	; Обнуление столбца COL.
23802	L1	11 3E 5D	LD DE, DATA2	; Данные для печати - DATA2.
23805		01 04 00	LD BC, 4	; Символов - 4.
23808		CD 3C 20	CALL 8252	; Печать строки DATA2.
23811	DELAY	21 FF FF	LD HL, 65535	Организация цикла
23814	L2	2B	DEC HL	задержки для того, чтобы
23815		7C	LD A, H	изменения на экране не
23816		B5	OR L	происходили слишком
23817		20 FB	JR NZ, L2	быстро.
23819		11 3E 5D	LD DE, DATA2	; Данные для печати - DATA2.
23822		01 04 00	LD BC, 4	; Символов - 4.
23825		CD 3C 20	CALL 8252	; Печать строки DATA2.
23828		3A 40 5D	LD A, (COL)	; Номер столбца позиции печати.
23831		3C	INC A	; Переход к соседней позиции.
23832		32 40 5D	LD (COL), A	; Запомнили новую позицию.
23835		ED 4B 3F 5D	LD BC, (LINE)	; Координаты позиции печати.
23839		CD 83 25	CALL 9603	; Вызов процедуры ПЗУ для опре-
				; деления атрибутов в текущей
				; координате.
23842		CD D5 2D	CALL 11733	; Перенос результата со стека
				; калькулятора в регистр A.
23845		FE 8F	CP 143	; Проверка на PAPER 1: INK 7:
				; FLASH 1.
23847		20 D1	JR NZ, L1	; Возврат, если нужно знако-
				; место еще не найдено.
23849		FD 36 57 00	LD (IY+87), 0	; Выключение режима INVERSE.
23853		C9	RET	; Возврат в БЕЙСИК.
23854	DATA1	DEFB 22 0 20 17		; Эти данные эквивалентны:
		1 16 7 18		; ...AT 0, 20; PAPER 1; INK 7;
		1 144 18 0		; FLASH 1; "A"; PAPER 0; INK 7;
		17 7 16 0		; FLASH 0. Здесь "A" - это сим-
				; вол UDG; размещенный на кла-
				; више A, т.е. CHR\$(144).
23870	DATA2	DEFB 22		; Эти данные эквивалентны:
23871	LINE	DEFB 0		; ...AT 0, COL; CHR\$(62).
23872	COL	DEFB 0		; Символ 62 - это ">", он и
23873	DEFB	62		; изображает "ракету".
23874	DATA3	DEFB 24 153 126 153		; Это данные для символа UDG,
		24 36 36 102		; изображающего человечка.

8.2 SCREEN\$

Точно так же, как и при вызове процедуры ATTR, перед вызовом этой процедуры необходимо в регистровую пару BC записать координаты символьной ячейки. Процедура

SCREEN\$ находится по адресу 9526. Параметры символа исследуемой символьной ячейки при вызове этой процедуры будут записываться на калькуляторный стек в пятибайтной форме. Использование калькуляторного стека для хранения параметров символа для Вас, наверное, открытие, поскольку до сих пор речь шла о хранении на стеке только чисел в пятибайтной форме.

Коды калькулятора при этом оперируют не с самими символами, а с их параметрами, записанными в пятибайтной форме. Для того, чтобы передать параметры символа на стек, необходимо записать в регистровую пару BC размер символа в байтах, а в регистровую пару DE - начальный адрес участка памяти, где хранится вся информация о символе. Для стандартных символов в регистр A записывается 0.

Для передачи этих параметров на стек используется процедура, находящаяся в ПЗУ по адресу 10929. Процедура же по адресу 11249 выполняет обратную функцию - передает параметры символа со стека в соответствующие регистры A, B, C, D, E. При этом размер символа будет записан в регистровую пару BC, а начальный адрес участка памяти с информацией о символе - в регистровую пару DE.

Программа 8.2 показывает, как можно использовать эти процедуры. После выполнения программы, на экране появятся два символа @, соответственно в позициях 0,20 и 0,21. Вместо вызова процедуры PRINT STRING (адрес 6252), можно использовать команду LD A,(DE) для проверки значения символа в позиции 0,20, а затем выполнить необходимые действия.

Листинг 8.2.

АДРЕС	МЕТКА	МАШ. КОД	АССЕМБЛЕР	КОММЕНТАРИЙ
			ORG 23750	
23760		3E 02	LD A, 2	; Канал экрана - 2.
23762		CD 01 16	CALL 5633	; Открываем канал печати на экран.
23765		CD 6B 0D	CALL 3435	; Очистка экрана.
23768		3E 02	LD A, 2	; Канал экрана - 2.
23770		CD 01 16	CALL 5633	; Открываем канал печати на экран.
23773		3E 16	LD A, 22	
23775		D7	RST 16	
23776		3E 0	LD A, 0	Аналог ...AT 0,20; "@"
23778		D7	RST 16	
23779		3E 14	LD A, 20	
23781		D7	RST 16	
23782		3E 40	LD A, 64	
23734		D7	RST 16	
23785		06 14	LD B, 20	; Номер экранного столбца.
23787		0E 00	LD C, 0	; Номер строки экрана.
23789		CD 38 25	CALL 9528	; Вызов процедуры ПЗУ для определения того, что находится в данном знакоместе.
23792		CD F1 2B	CALL 11249	; Передача параметров символа со стека калькулятора в регистры процессора.
23795		CD 3C 20	CALL 8252	; Печать той символьной строки, на местоположение которой указывает пара DE, а длина которой находится в BC. В таком случае - это символ "@".
23798	C9		RET	

При использовании БЕЙСИК-оператора SCREEN\$ имеет место одно ограничение. Код распознаваемого символа на экране должен быть в диапазоне от 32 до 127. В противном случае считается, что это пустой символ. Такое ограничение существенно для игровых программ, где используются UDG, распознать которые в данном случае не удастся. Для программ в машинных кодах такой проблемы не существует, поскольку UDG можно включить в стандартный символьный набор. Постоянно этот набор хранится в ПЗУ, начиная

с адреса 15360. Однако начальный адрес этого набора можно менять, меняя значение системной переменной CHARS (23606/7). Она указывает на 256 байтов ниже, чем начало символьного набора. Если переписать символьный набор из ПЗУ в ОЗУ таким образом, чтобы в этот набор вышел участок с информацией об UDG и чтобы UDG "A" соответствовал код 32, UDG "B" - код 33 и т.д., то можно вызывать процедуру SCREEN\$ для идентификации UDG на экране.

Программа 8.3 работает точно так же, как и программа 8.1, но эта программа способна распознавать на экране символ, код которого равен 144. Выполнив копирование стандартного символьного набора в ОЗУ и переписав значение символьной переменной CHARS можно совсем отказаться от UDG. Теперь можно часть стандартных символов (например, строчные буквы и некоторые знаки пунктуации) заменить на символы, форму которых можем задать сами (это будет около 50 символов). Процедура SCREEN\$ будет легко распознавать на экране любой символ, если он входит в символьный набор. И еще одно важное замечание. Необходимо всегда иметь свободный участок памяти, используемый как рабочий для стека. Если такого участка нет или он есть, но периодически не очищается, то при выполнении программы может появиться сообщение о переполнении памяти. Поэтому перед каждым вызовом процедуры SCREEN\$ необходимо очищать специально зарезервированный рабочий участок.

Листинг 8.3.

АДРЕС	МЕТКА	МАШ. КОД	АССЕМБЛЕР	КОММЕНТАРИЙ
23760				
.....	СМ.	ЛИСТИНГ 8.1		
			ORG 23760	
23833				
23836	2A	7B 5C	LD HL, (23675)	; Адрес первого символа UDG.
23839	25		DEC H	; Отступили на 256 байтов.
				; Мы помним, что CHARS указы-
				; вает на 256 байтов ниже, чем
				; начало шрифта.
23840	22	36 5C	LD (23606), HL	; Переустановили CHARS так,
				; она теперь указывает на
				; область UDG.
23843	CD	38 25	CALL 9528	; Вызов процедуры SCREEN\$.
23846	CD	F1 2B	CALL 11249	; Передача параметров симво-
				; ла со стека в регистры.
23849	1A		LD A, (DE)	; Проверка номера символа
				; в текущем знакоместе.
23850	FE	20	CP 32	; Проверка "не пробел ли это?"
23852	21	00 3C	LD HL, 15360	; Расположение стандартного
				; шрифта минус 256 б.
23855	22	36 5C	LD (23606), HL	; Восстановили стандартное
				; значение CHARS.
23858				

..... Далее, как в программе 8. 1

Процедура очистки рабочего участка имеется в ПЗУ по адресу 5823. Вызов этой процедуры в программе 8.3 можно производить после команды LD A,(DE), однако предварительно необходимо сохранить в памяти последнее значение регистра A.

8.3 POINT

Оператор POINT x,y в БЕЙСИК-программах выдает 1, если пиксел экрана с координатами x,y включен и выдает 0, если он выключен. Процедура ПЗУ, соответствующая этому оператору (адрес 8910), может быть вызвана, если предварительно записать в регистр B y-координату, а в регистр C, соответственно x координату пиксела. При этом результат зашлется на стек, а т.к. он равен 0 или 1, то мы можем переписать его в регистр A и сравнить с заданным значением. В программе 8.4 показан способ проверки пиксела верхнего левого угла (координаты 0,175).

Если в начале в символьную ячейку с координатами 0,0 занести черный квадрат, то

результат работы процедуры будет равен 1, т.к. пиксел с координатами 0,175 входит в этот квадрат.

Листинг 8.4.

АДРЕС	МЕТКА	МАШ. КОД	АССЕМБЛЕР	КОММЕНТАРИЙ
23760		3E 02	LD A, 2	; Канал экрана - 2.
23752		CD 01 16	CALL 5633	; Открываем канал печати на экран.
23765		CD 6B 0D	CALL 3435	; Очистка экрана.
23768		3E 02	LD A, 2	; Канал экрана - 2.
23770		CD 01 16	CALL 5633	; Открываем канал печати на экран.
23773		3E 8F	LD A, 143	; "Черный квадрат".
23775		D7	RST 16	; Печать квадрата.
23776		0E 00	LD C, 0	; Координата x.
23778		06 AF	LD B, 175	; Координата y.
23780		CD CE 22	CALL 8910	; Вызов процедуры POINT из ПЗУ.
23783		CD D5 2D	CALL 11733	; Перенос результата со стека калькулятора в регистры.
23786		FE 01	CP 1	; Проверка включен пиксел или нет.
23788		C8	RET Z	; Выход, если да.

9. ПРИНТЕР

Не так уж много существует программ, для работы которых необходим принтер. Всегда можно ограничиться для вывода информации только экраном дисплея. Но иногда, например, необходимо показать свой уровень, достигнутый в какой-нибудь игре друзьям или знакомым. В этом случае нужно сделать копию изображения экрана на принтере. Поэтому в этой главе мы рассмотрим три BASIC-команды: COPY, LPRINT, LLIST и их аналоги в машинных кодах, обеспечивающих работу с принтером.

9.1. COPY

При вызове этой процедуры только двадцать две строки экрана передаются на принтер. Программа в машинных кодах позволит Вам улучшить эту процедуру так, чтобы можно было передавать на принтер произвольное число строк, причем, начиная с любой строки на экране, например, передать на принтер три строки, начиная с седьмой.

В программе 9.1 показан способ передачи на принтер всех 24 строк экрана, поскольку для некоторых игр достигнутый уровень показан в нижних строках экрана и при обычном вызове процедуры COPY получить необходимую информацию с двух последних строк не удастся.

Листинг 9.1

		ORG 23760
23760	21 00 40	LD HL, 16384
23763	06 C0	LD B, 192
23765	F3	DI
23756	CD B2 0E	CALL 3762
23769	C9	RET

Как Вы заметили, в этой программе перед вызовом процедуры, находящейся в ПЗУ по адресу 3762, необходимо в регистровую пару HL записать начальный адрес экранной области ОЗУ (т.к. нам необходима копия, начиная с первой строки, то этот адрес 16384), а в регистр B - число копируемых строк, умноженное на 8.

9.2 LPRINT

Эта команда имеет сходство с командой PRINT. Различие состоит в том, что не работают по этой команде управляющие коды и, конечно, используется другой канал для вывода информации на принтер, который необходимо открыть в программе в машинных

кодах, прежде чем использовать команду RST 16. Например, чтобы вывести на принтер литеру "A", т.е. реализовать в машинных кодах LPRINT "A", необходимо первоначально открыть канал 3, затем записать в регистр A код литеры "A", ну а затем использовать команду RST 16.

Программа 9.2 демонстрирует этот способ для вывода на принтер символьной строки.

Листинг 9.2

```
                ORG 23760
23760           3E 03      LD A, 3
23762           CD 01 16   CALL 5633
23765           11 DF 5C   LD DE, DATA
23768           01 09 00   LD BC, 9
23771           CD 3C 20   CALL 8252
23774           C9        RET
23775 DATA DEFM PROGRAM 2
```

Для выполнения табуляции выводимой на принтер информации, необходимо предварительно в системную переменную PRCC (23680) через POKE записать соответствующее число.

9.3 LLIST

Это третья БЕЙСИК-команда для работы с принтером. Реализовать ее в машинных кодах довольно просто. Достаточно только записать CALL 6133 и будет выполняться листинг строк BASIC-программы на принтере.

СТРАНИЦА iS-DOS

(C) Самыловский С.В., 1993.

(C) SLOT Co.Ltd., Moscow, 1993.

Новинка сезона - дисковая операционная система для SPECTRUM "iS-DOS"

Еще несколько месяцев назад это казалось невероятным. Разрыв между желанием иметь дома профессиональный персональный компьютер и возможностью купить его становился все больше и больше. Красавцы PC достойно заняли свое место в институтах, банках, офисах и пока очень редко перешагивают порог нашего дома. Да и нужен ли каждому из нас дома, в школьном классе, в офисе небольшой частной фирмы супер-компьютер типа IBM PC, стоимостью более миллиона рублей? Зададим себе вопрос: что бы мы хотели поручить своему электронному помощнику? При всем кажущемся многообразии ответов на данный вопрос, все они могут быть сведены к следующим:

- игры и развлечения для детей и взрослых;
- подготовка, хранение и печать текстов и документов;
- выполнение текущих расчетов, решение конкретных задач;
- обучение, тренировка, управление какими-либо устройствами;
- профессиональное использование компьютера, составление программ на различных языках программирования, их отладка и т.п.;
- коммуникационные функции - домашний секретарь, телефонный автоответчик, прием и передача текстовых и графических документов по телефонным линиям связи.

Итак, сможет ли каждый из нас в обозримом будущем иметь у себя дома такой компьютер? Посмотрите внимательно на свой SPECTRUM! Если у него предусмотрена возможность подключения накопителя на гибком магнитном диске (дисквода) и печатающего устройства (принтера), то уже завтра Вы можете стать обладателем вычислительной системы, которая решит большинство Ваших домашних проблем, поможет в учебе и работе, позволит в спокойной домашней обстановке приобщиться к такому загадочному и такому интересному миру настоящих компьютерных технологий! Это маленькое чудо сделает для Вас и Вашего SPECTRUM-совместимого компьютера новая дисковая операционная система iS-DOS!

О чем собственно речь?

Речь идет об открытой многокаталоговой операционной системе iS-DOS, результате многолетнего труда коллектива разработчиков из Санкт-Петербургской фирмы ISKRASOFT, под руководством Ильяшенко Э.Н. Само название операционной системы iS-DOS - это сокращение от IskraSoft Disk Operating System. Оно, как и товарный знак системы закреплено законодательно за данной фирмой и без ее согласия не может быть использовано кем либо в коммерческих и рекламных целях.

На Российском рынке iS-DOS появилась в конце 1992 года и с этого момента уже сумела завоевать многочисленные симпатии пользователей SPECTRUM-совместимых компьютеров. Подробнее об истории создания системы и основных идеях, положенных в ее основу мы расскажем на страницах первых выпусков "ZX-РЕВЮ-94".

Компьютер, который мы выбираем.

Важнейшим достоинством системы iS-DOS является ее высокая мобильность и "неприхотливость" в требованиях к компьютеру. Она будет работать на любом SPECTRUM-

совместимом компьютере, имеющем в своем составе хотя бы один накопитель на гибком магнитном диске (дискет). Для работы iS-DOS требуется всего 48 Кбайт оперативной памяти. Остальная имеющаяся в компьютере оперативная память может быть использована для буферизации ввода-вывода или создания электронного диска, что примерно на порядок может повысить быстродействие работающих программ. iS-DOS обеспечивает использование до 4-х накопителей на гибких магнитных дисках, электронного диска и одного жесткого диска типа винчестер одновременно. Работа с различными клавиатурами от обычной 40-клавишной до XT-клавиатуры поддерживается набором специальных драйверов клавиатуры.

К Вашему компьютеру подключен принтер? В богатом перечне драйверов принтеров Вы наверняка быстро найдете нужный. Если принтер обеспечивает печать символов псевдографики, то высокое качество печатаемых документов, таблиц, графиков Вам обеспечено. Если принтер старенький то iS-DOS учтет и это!

Имея "ZX-модем" в составе компьютера и установив iS-DOS, Вы легко сможете связаться по телефону со своим коллегой и передать ему несколько файлов, а если Ваш компьютер обеспечивает последовательный интерфейс RS-232, то подключив к нему стандартный для IBM PC модем, Вы, с помощью специальной коммуникационной программы сможете выйти в настоящие компьютерные сети.

Работа с цветным монитором несомненно доставит Вам удовольствие от общения с системой. Если Ваш компьютер подключен к цветному или черно-белому телевизору, не переживайте! Вы можете работать в режиме 41 или 61 символ в строке, а возможность настройки цвета используемых программ сделает работу с компьютером максимально удобной.

Итак, все зависит только от наших возможностей, а начать можно и с малого! В любом случае возникает вопрос:

Так где же взять iS-DOS систему?

Кому-нибудь это может показаться странным, но iS-DOS придется купить. Переписать ее у товарища не удастся, так как она неплохо защищена от копирования и в лучшем случае "не захочет" загрузиться. Не лучшим вариантом будет и покупка системы с рук у знакомого или (что гораздо хуже) у незнакомого умельца хэккера. Ряд интересных программ вероятнее всего не смогут работать корректно или вообще не будут запускаться, а это так обидно...

Официально приобрести iS-DOS можно у самой фирмы ISKRASOFT в С. Петербурге, у ее Генерального дистрибьютора - московского предприятия "Slot" Co.,Ltd или у их официальных дилеров в Вашем городе или области. Заказав систему, можно также получить каталог прикладных программ и литературы по iS-DOS почтой по адресам:

117330 Москва, а/я 707, фирма "Slot"

или

121019 Москва, а/я 16, фирма "ИНФОРКОМ".

В Москве iS-DOS, различные прикладные программы для нее и документацию Вам предложит фирма "Slot", контактный телефон (095) 143-11-91, в следующих местах:

- на корпункте "ИНФОРКОМА" по адресу: Москва, Новый Арбат д.2, 19-е отделение связи (1-й этаж операционного зала);
- в здании Политехнического музея в отделе "Вычислительная техника" по адресу: Москва, Новая пл., д.3/4 п.1, 3-й эт.
- на московском радиорынке в выходные дни у официальных дилеров фирмы "Slot".

Система и все прикладные программы к ней доставляются на качественных дискетах, в фирменной упаковке фирмы "Slot", со специальной фирменной маркировкой и комплектуются кратким описанием покупаемого продукта, необходимого для начала работы HELPa и "Регистрационной карточки пользователя" которая дает ее обладателю право на 50% скидку при покупке новой версии программы и бесплатное использование "горячей

линии помощи".

Зарегистрированным пользователям фирма в первую очередь предоставляет информацию о новых поступлениях программ и документации, проводит с ними индивидуальные консультации, организует обучение и семинары по обмену опытом.

Первое знакомство с iS-DOS.

iS-DOS выгодно отличается от существующие дисковых операционных систем для компьютеров семейства "ZX-SPECTRUM" и обладает более широкими возможностями, чем, например, традиционная TRDOS. При разработке ОС iS-DOS использованы стиль и идеология известной операционной системы MS-DOS и оболочки "Norton Commander" для ЭВМ IBM PC, а также стремление к максимальному удобству для пользователя. Такой подход к созданию системы обладает определенным преимуществом - пользователь, работающий на ПЭВМ типа "ZX-SPECTRUM" с iS-DOS при переходе на IBM PC попадает в знакомую, привычную уже среду "NORTON COMMANDER".

Операционная система iS-DOS позволяет работать с разнообразными внешними устройствами, такими, как электронный диск, дисковод 5.25 дюйма, винчестер, модем и локальная сеть, принтеры различных типов.

Полноценный текстовый редактор "Editor", программы печати текстовых и графических файлов, поддержка модема превращают обыкновенный "ZX-Spectrum" в полноценное рабочее место современного делового человека. iS-DOS значительно расширяет возможности компьютеров типа "ZX-Spectrum" для использования их в научной деятельности, в сфере бизнеса и управления, для создания прикладных и коммерческих программных продуктов.

iS-DOS ориентирована на пользователя, не обладающего навыками программирования, но не оставит равнодушным и профессионального программиста. И первое, с чем Вам предстоит познакомиться - это файловая оболочка системы.

ФАЙЛОВАЯ ОБОЛОЧКА. (SHELL)

Файловая оболочка iS-DOS предназначена для хранения, просмотра, редактирования и запуска файлов. В нее Вы попадаете после запуска операционной системы. Для этого необходимо вставить дискету с ос iS-DCS в дисковод и нажать клавишу <Reset>, т.е. перезапустить компьютер. Загрузчик iS-DOS перехватывает обращение TRDOS к диску и передает управление iS-DOS. После загрузки Вы увидите две панели с содержимым Вашего диска, а верхняя строка показывает системные утилиты, которыми Вы можете пользоваться при работе в оболочке. Эти утилиты вызываются цифровыми клавишами <1>, ..., <0> соответственно и выполняют следующие функция:

- | | |
|--------|---|
| HELP | - справочная информация. |
| USER | - вызов пользовательского меню. |
| VIEW | - просмотр файла. |
| EDIT | - вызов текстового редактора. |
| COPY | - копирование файлов. |
| RENAME | - переименование файлов, каталогов. |
| crDIR | - создание подкаталога. |
| DELETE | - удаление файлов, каталогов. |
| MENU | - системное меню. |
| MASK | - установка имен, расширений, параметров, файлов для отображения на |

панели.

На верхней строке правой и левой панели указано текущее устройство и имя диска или подкаталога. На правой и левой панели - содержимое главного каталога. Чтобы оказаться в нужном Вам подкаталоге подведите курсор к нему и нажмите <Enter>. Движение курсора осуществляется клавишами:

- <A> - вниз

- <Q> - вверх
- <O> - на левую панель
- <P> - на правую панель

Здесь работают также стандартные клавиши управления курсором.

- <CS/Q> - в начало каталога.
- <CS/A> - в конец каталога.
- <CS/O> - в корневой каталог левой панели.
- <CS/P> - в корневой каталог правой панели.
- <SS/CS>- обмен панелей.

Выбор нового устройства:

- <CS/1>- на левой панели.
- <CS/2>- на правой панели.

При выборе нового устройства появится меню:

Choose Drive:						
A	B	C	D	E	F	

Управляя курсором клавишами <O>, <P> можно выбрать новое устройство с помощью нажатия <Enter>. Отказаться от выбора и выйти в оболочку можно с помощью <SS/A> или <Space>. Подсказки по программам iS-DOS вызываются с помощью клавиши <1>. Для этого необходимо подвести курсор к интересующей Вас программе и нажать <1>. Для просмотра текстового файла подведите курсор к нему и нажмите клавишу <3>. Для управления оболочкой задействованы два типа клавиш - зарезервированные для оболочки и переопределяемые пользователем.

К первому типу относятся: клавиши <O>, <P>, <A>, <Q> и их комбинации с <CS>, клавиш <5>, <6>, <7>, <8>, <9> и комбинации <CS/1> <CS/2>, а также <Enter> и <Space>. Клавиши <O>, <P>, <A>, <Q>, а также <5>, <6>, <7>, <8> в сочетании с <CS> - управляют движением курсора (такое управление является стандартным для iS-DOS).

Второй тип клавиш и их функции заданы в файле extkey.txt, в котором Вы можете сами определить функции клавиш так, как Вам удобно. По умолчанию поддерживаются следующие определения (один из вариантов):

- c - сравнение файлов на панелях;
- C - калькулятор;
- SS+c - копировщик каталогов;
- d - запрос даты;
- D - вывод данных о файле на экран;
- e - просмотр и удаление резидентив;
- f - поиск файла;
- F - копирование файла с сегментацией;
- h - распечатка текста или картинки;
- l - свободное место на диске;
- m - монитор командной строки;
- M - резидентный монитор ком. строки (необходимо сначала установить);
- SS+k - маркировать все файлы в каталоге (+);
- SS+j - отменить маркировку всех файлов (-);
- n - копирование файла с удалением (move);
- r - удаление целого каталога с файлами;
- S - вывод системной информации;
- s - сортировка по имени файла;
- SS+s - сортировка по расширению файла;

- t - вывод 'дерева' каталогов;
- u - восстановление удаленных файлов.

Начальная загрузка системы.

При начальной загрузке операционная система ищет в корневом каталоге диска, с которого она загружена, файл autoexec.bat. Если этот файл будет найден, то он выполняется.

Файл autoexec.bat содержит команды, которые должны выполняться каждый раз при начальной загрузке системы. Эти команды осуществляют необходимую настройку операционной системы, устанавливают удобное для работы окружение, запрашивают текущую дату, устанавливают список каталогов в которых производится поиск выполняемых программ (команда path), устанавливают текущий каталог рабочей панели и выполняют другие полезные действия.

ПОДПРОГРАММА ВЫЗОВА СПРАВОЧНОЙ ИНФОРМАЦИИ (ПОДСКАЗОК) (HELP)

Программа вызова справочной информации в основном меню системы (фигурирует как " HELP ") вызывается клавишей <1>.

Для получения справочной или дополнительной информации по интересующей Вас программе найдите файл этой программы курсором и нажмите <1>. Что Вы увидите? Это зависит от содержания файла с именем интересующей Вас программы, но с расширением .hlp, находящегося в каталоге HELP. Если Вы увидите:

" Welcome to Shell for iS-DOS-92 ... "

то это означает, что в каталоге HELP отсутствует соответствующий файл с расширением .hlp и Вам предложена справка по оболочке.

МЕНЮ ПОЛЬЗОВАТЕЛЯ (USER)

Программа предназначена для удобства работа пользователя в SHELL (аналог программирования реакции на нажатие клавиши с большими возможностями, удобна для использования при быстрых переходах из каталога в каталог, для часто вызываемых программ, для вызова прикладных пакетов и т.д.). Программа вызывается клавишей <2>.

Для работы программы необходим текстовый файл menu.txt. При вызове программа вначале ищет файл menu.txt в текущем каталоге, затем (при его отсутствии там) в каталоге SHELL, таким образом пользователь может иметь в каждом каталоге свое меню.

ПРОГРАММА ПРОСМОТРА ТЕКСТОВЫХ ФАЙЛОВ (VIEW)

Для просмотра текстового файла удобно воспользоваться функцией VIEW. Для этого подведите к нему курсор и назовите клавишу <3>. На экране появится больше окно, в котором распечатается содержимое файла. По файлу можно двигаться с помощью клавиш управления курсором, а также в режиме непрерывного скроллинга. Можно переключить режим отображения текста на экран: 41/61 буква в строке.

ТЕКСТОВЫЙ РЕДАКТОР (EDIT)

Мощный и удобный в работе текстовый редактор EDIT несомненно придется по душе даже профессионалу. По своим возможностям он ненамного уступает знаменитому текстовому редактору LEXICON для IBM PC. Кроме обычных функций удаления, вставки и построчного редактирования в нем реализованы возможности контекстного поиска и замены, блочные операции над текстом, режим использования псевдографики, разбивка

текста на страницы, автоматическое или построчное форматирование текстовых абзацев с выполнением переносов слов по правилам орфографии, возможность одновременной работы с несколькими текстовыми файлами, работа с макросами.

Для вызова текстового редактора подведите курсор к файлу, который Вы желаете отредактировать, и нажмите клавишу <4>. Если тип выбранного Вами файла относится к допустимому, то редактор предложит Вам его отредактировать, создать новый файл, или отказаться и от того и от другого. Вам будет также предоставлена возможность копировать файл в резервный файл расширением .bak. В противном случае Вам будет предложено лишь создать новый файл.

КОПИРОВАНИЕ ФАЙЛОВ (COPY)

Удобно и понятно в системе реализована функция копирования файлов. Для того, чтобы скопировать файл, надо подвести к нему курсор и нажать клавишу <5>. Для копирования группы файлов их надо предварительно выделить с помощью клавиши пробел. Копирование всегда происходит с текущей панели на противоположную. Включение режима работы с буферизацией заметно ускорит выполнение больших операций копирования.

РЕДАКТИРОВАНИЕ АТТРИБУТОВ ФАЙЛА ИЛИ КАТАЛОГА (RENAME)

Изменить имя файла или его расширение поможет Вам функция RENAME. Она работает с атрибутами как файла, так и каталога. Установив курсор на файл, с которым Вы желаете работать, нажмите клавишу <6>. При этом загорится 6-ий слева индикатор "RENAME" и возле выбранного Вами файла, замигает курсор строкового редактора. Изменив имя или расширение файла нажмите ENTER.

СОЗДАНИЕ НОВОГО КАТАЛОГА (MKDIR)

Для создания нового каталога (подкаталога) нажмите клавишу <7> MKDIR. В ответ на появившийся запрос введите имя создаваемого каталога.

УДАЛЕНИЕ ФАЙЛОВ и КАТАЛОГОВ (DELETE)

Удаление файлов или пустых каталогов выполняется клавишей <8>. Подведите курсор к удаляемому файлу и нажмите <8>. Для удаления группы файлов их надо предварительно выделить с помощью клавиши <пробел>. После ввода подтверждения файлы будут уничтожены.

СИСТЕМНОЕ МЕНЮ (MENU) ВВОД МАСКИ-ШАБЛОНА ФАЙЛА. УСТАНОВКА РЕЖИМОВ ПЕЧАТИ КАТАЛОГА (MASKA)

Две последние функции файловой оболочки MENU <9> и MASKA <0> помогут Вам настроить систему для быстрого поиска и удобной индикации в текущем каталоге файлов с необходимыми именами и (или) расширениями.

РАБОТА С TR-DOS ДИСКАМИ В СРЕДЕ IS-DOS.

Серьезное внимание разработчики системы уделили вопросам копирования информации из TRDOS в iS-DOS и обратно. Копирование файлов с TRDOS-дисков выполняется программой FROM_TRD и запускается клавишей <f> или из "USER"-меню. Текстовые файлы TLW с помощью программы FROM_TLW после копирования можно преобразовать в формат редактора EDIT системы iS-DOS.

Копирование на TRDOS-диски выполняется программой TO_TRDOS и вызывается клавишей <t> или из "USER"-меню. Отметив копируемые файлы пробелом или подведя

курсор к копируемому файлу нажмите <t>. Вам будет предложено меню, в котором можно отказаться от копирования, изменить текущие параметры копирования или выполнить его, нажав <ENTER>.

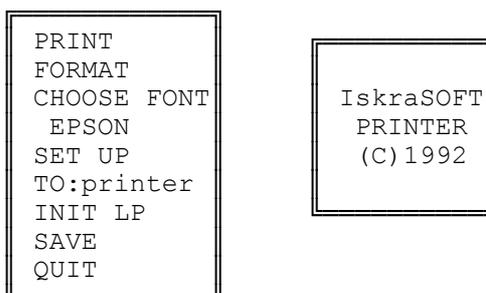
РАБОТА С MS-DOS ДИСКАМИ В СРЕДЕ iS-DOS.

Работая в iS-DOS, Вы получаете возможность прочитать дискету в формате MS-DOS емкостью 360к, просмотреть каталоги и подкаталоги на ней, удалить файлы и каталоги, переименовать файлы и каталоги, создать новые каталоги, скопировать файл или группу файлов на диск iS-DOS и обратно, перекодировать скопированный из MS-DOS текстовый файл в формат редактора EDIT системы iS-DOS и выполнить обратную операцию.

Обращает на себя внимание отсутствие со стороны iS-DOS ограничения на размер обрабатываемых файлов MS-DOS. Это наверняка заинтересует тех, кто хотел бы результаты домашней работы перенести потом на более мощный компьютер для загрузки в базу данных или распечатки на лазерном принтере.

ПРОГРАММА ПЕЧАТИ ТЕКСТОВЫХ ФАЙЛОВ.

Программа печати текстовых файлов в iS-DOS выполнена в виде отдельной программы PRINT, которая вместе с редактором EDIT позволит Вам получить качественный документ, используя для этого все возможности подключенного к компьютеру принтера. Для вызова программы печати (предварительно загрузив драйвер печати) установите курсор на текстовом файле и нажмите клавишу <H>. Загорится индикатор "VIEW", и Вы окажетесь в меню программы печати текстовых файлов iS-DOS:



До начала печати можно определить формат печатаемого документа, количество столбцов печати, необходимость отступов, нумерации листов, плотность печатаемых символов, качество шрифта и другие параметры. Для настройки на конкретный тип принтера имеется специальное меню, которое предлагает Вам на выбор: EPSON, MC6312, MC6313, MC6337. Если в этом перечне Вы не нашли подходящий для себя принтер, имеется возможность описать в специальном файле свой принтер самостоятельно.

ПРОГРАММА ПЕЧАТИ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ И РИСУНКОВ ИЗ ФАЙЛОВ ТИПА *.scr.

Программа печати графических изображений практически эмулирует возможности известного графического пакета STORY BOARD FOR MS-DOS. Возможность масштабирования изображения, режим ЛУПА, свободное размещение на листе, печать в полутонах полностью удовлетворят Ваши потребности.

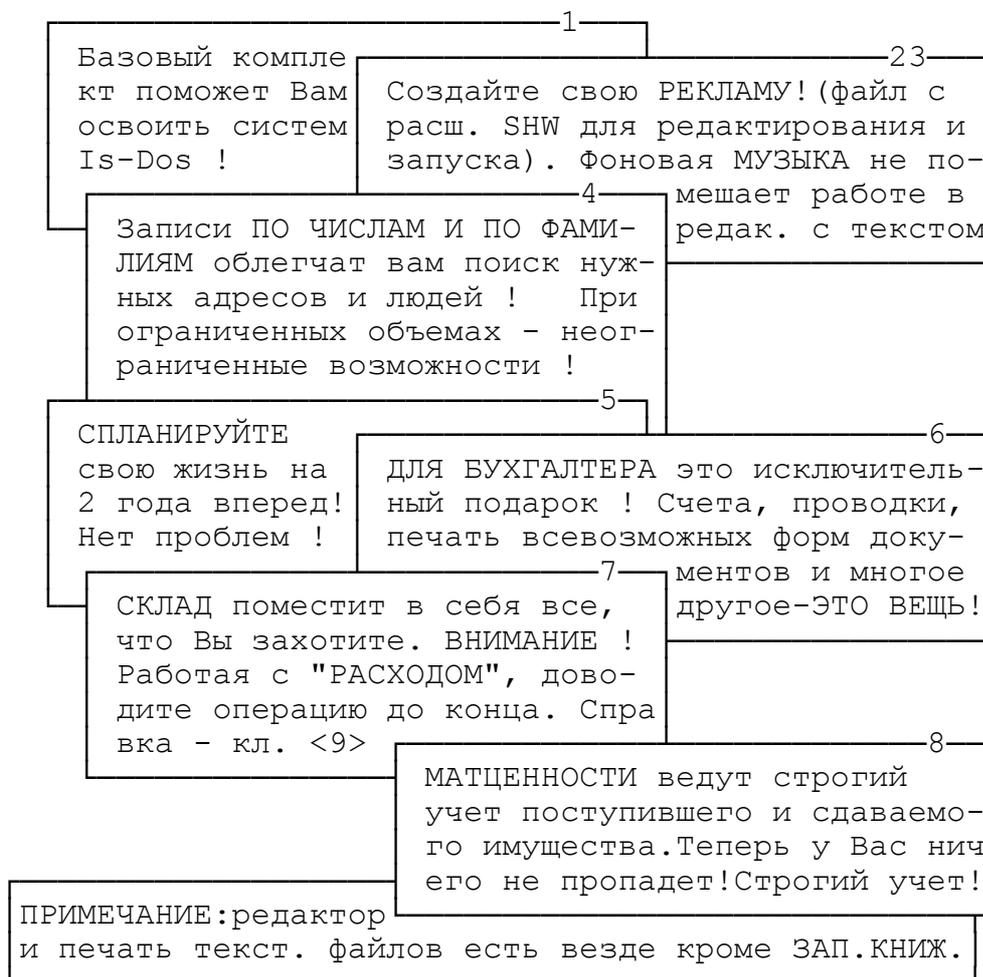
Перед запуском программы должен быть инсталлирован драйвер Вашего принтера. Допустимый тип печатаемых файлов *.scr. При переброске графических файлов из TRDOS, не забывайте их переименовывать. Далее подведите курсор к файлу с расширением scr, из которого Вы желаете распечатать графические изображения и нажмите клавишу H.

Продолжение следует.

КОМПЛЕКТ ДИСКЕТ Is-Dos
поставляемый фирмой "Slot" г. Москва

N	НАИМЕНОВАНИЕ	СОДЕРЖАНИЕ
01	БАЗОВЫЙ КОМПЛЕКТ	ОС is-DOS; Утилиты; MS-Dos; Текстовый редактор; Программа печати текстовых файлов;
23	PRINTLUX & MUSIC RECLAM SHOW	ОС is-Dos; Печать; Редактор; Программа печати экранных файлов + .SCR файлы; Пакет фонового музыкального сопровождения на базе AY-процес. Программа создания динамических рекламных роликов;
04	ЗАПИСНАЯ КНИЖКА	ОС is-Dos; Пакет Записная книжка"
05	ДЕЛОВОЙ КАЛЕНДАРЬ	ОС is-Dos; Печать; Редактор; Пакет "Деловой календарь"
06	АРМ "ФИНАНСЫ БЕЗ ПРОБЛЕМ"	ОС is-Dos; Печать; Редактор; Пакет"Финансы без проблем"
07	АРМ "СКЛАД"	ОС is-Dos; Печать; Редактор; Пакет "Склад"; "Настройщик"
08	АРМ "МАТЕРИАЛЬНЫЕ ЦЕННОСТИ"	ОС is-Dos; Печать; Редактор; Пакет "Матценности"

Distributed by SLOT Co.Ltd Moscow Tel:(095)143-11-91



Комментарий ИНФОРКОМа.

При первом взгляде на то, о чем Вам сегодня рассказали, у пользователя возникает обычно два вопроса:

- НЕУЖЕЛИ ЭТО СДЕЛАНО У НАС?
- ПОЧЕМУ Я ОБ ЭТОМ ЕЩЕ НИЧЕГО НЕ ЗНАЮ?

То же произошло и с нами. Где-то начиная с июля месяца 1993 года в нашей почте все чаще стали проскакивать сообщения о том, что где-то кто-то что-то видел или слышал. Люди просили рассказать, что мы знаем об iS-DOS.

Комментарии, которыми сопровождалось подобные сообщения, обычно соответствовали тем, которыми сопровождают сообщения об НЛО. Основная суть: это невероятно, этого просто не может быть, потому что этого не может быть никогда.

Мы начали "раскручивать" этот вопрос. Сначала приобрели пару систем и проверили их в деле. Оказалось, что слухи не вполне правильно описывали действительность. **ДЕЙСТВИТЕЛЬНОСТЬ ОКАЗАЛАСЬ ЕЩЕ НЕВЕРОЯТНЕЕ.**

И сейчас мы делаем однозначный вывод: "Господа, пока Вы читаете эти строки, Ваш компьютер переживает второе рождение, а у Вас начинается новая жизнь! О том, что сделали Ваши коллеги, российские программисты, К. Синклер даже и не мечтал.

Да, это действительно сделано у нас в России! И мы еще расскажем на страницах "ZX-РЕВЮ" о том, как это было сделано и расскажем о тех людях, которые стояли у истоков новой системы.

Самое же потрясающее, что это действительно многолетний труд, который не вышел за стены лаборатории до тех пор, пока система не была полностью подготовлена и обкатана и пока она не "обросла" необходимым пакетом утилит, драйверов и прикладных программ, то есть всем тем, без чего ее нельзя было бы взять такую, как она есть и сразу использовать на полную мощность. Это уникальный пример грамотного, но очень трудного с инженерной и экономической точки зрения маркетинга. Вы только представьте, через что пришлось пройти разработчикам, которые несколько лет работали в поисках и сомнениях, не имея возможности ни оценить спрос на свой труд, ни получить за него минимальную компенсацию. И если коллектив выжил в таких условиях, довел свое дело до конца, то ему просто надо ставить памятник при жизни.

Из статьи руководителя фирмы "SLOT", Самыловского Сергея Владимировича Вы узнали самые первые сведения об iS-DOS. По должности автор говорит в этой статье о том, что сейчас есть реально и что может быть взято Вами немедленно.

У нас же руки в этом смысле развязаны и мы можем вместе с Вами, уважаемые читатели, заглянуть в ближайшее будущее.

Система открыта, а это значит, что она представляет неограниченный простор для творчества системных программистов, которые могут уже сейчас работать над ее развитием, над разработкой новых системных утилит и драйверов любых внешних устройств. Основное достоинство состоит в том, что каждый, кто способен сделать нечто значимое, сделает это не для корзины и не для узкого круга близких друзей, а для тысяч счастливых обладателей новой системы и его труд получит достойное моральное и материальное вознаграждение.

Программисты-прикладники, опирающиеся на возможности новой системы, получают необходимую поддержку от аппаратной среды (несколько дисководов, винчестер, средства коммуникации и т.д. и т.п.). Они смогут не просто работать, а работать "под систему" и быть уверенными, что их труд не останется невостребованным.

Поддержка из системы iS-DOS сетевых возможностей даст неограниченный простор для творчества всем желающим. Это и возможность работы с BBS и обмен информацией через модем. Возможно, уже в 1994 году в разных уголках страны появятся электронные журналы по "Синклеру". Каждый желающий сможет стать и автором и главным редактором и издателем.

Совместимость файлов с форматом MSDOS фактически превращает Ваш "Спектрум" в IBM-совместимую машину (на уровне системы). Работая дома, по сети Вы получите доступ к ресурсам IBM-компьютера, установленного у Вас на работе. Вы сможете по-новому организовать свой рабочий день и перейти к тому стилю, который уже давно практикуется в развитых странах.

Ресурсы IBM-совместимой техники смогут быть использованы на благо "Спектрума". Появятся редакторы и АССЕМБЛЕРЫ для "Спектрума", работающие на IBM. Открывается возможность создания программ КРОСС-ассемблеров для перевода машинного кода программ, написанных для одного компьютера на машинный код другого. Все это возможно, но требует больших ресурсов памяти и большого дискового пространства для размещения функциональных библиотек. Теперь эти возможности начнут приходить в Ваш дом.

Об уникальных возможностях использования "Спектрума" в качестве офисного оборудования мы уже и не говорим, они очевидны.

Новая система предоставляет необычные возможности для заводов-изготовителей компьютеров. В своих производственных планах они уже сейчас могут учесть включение в принципиальную схему ПК узлов, которые позволят взять от машины максимум возможного под управлением iS-DOS. Не исключено, что будут заключены лицензионные соглашения между заводами и авторами системы о включении в архитектуру компьютеров ПЗУ с системой iS-DOS.

Все это перспективы ближайших нескольких месяцев. А что же для этого делается сейчас?

Между фирмами "ИНФОРКОМ" и "SLOT" заключено соглашение о совместных действиях по продвижению системы iS-DOS на рынок СНГ. Отныне, начиная с января 1994 г. "ИНФОРКОМ" начинает эксклюзивное освещение новой системы. Принято решение о пятикратном увеличении тиражности "ZX-РЕВИЮ" в 1994 году и о существенном улучшении его полиграфического исполнения. Мы надеемся, что все желающие получать это издание наконец-то смогут это сделать и не остаться за бортом стремительного распространения iS-DOS.

Программа действий согласована с авторами системы, фирмой Iskra-Soft (С.Петербург) и получила полную поддержку.

Предполагается, что по мере распространения системы и завоевания ею популярности, будет открываться и доступ к ее работе изнутри (для системных программистов). Со временем будут опубликованы точки входа в систему и закрытая часть информации о ее работе, короче говоря, распространение системы будет сопровождаться ее "открыванием" в расчете на то, что ее распространение будет поддержано дальнейшим ее развитием. Возможно, будет создана сертификационная группа для включения в систему разработок сторонних специалистов.

Подготовлена и сдана в печать первая книга об iS-DOS, выход из печати ожидается в конце ноября - начале декабря 1993 года. Книгу можно будет и приобрести как в фирме "SLOT", так и по каналам "ИНФОРКОМа".

Одним словом, всех нас ждут в 1994 году большие события. Настраивайтесь на серьезную работу, будущее обещает быть интересным.

Ваш "ИНФОРКОМ".

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

(С) Збитнев В.А., г. Новосибирск, 1993г.

КОМПЬЮТЕР И ЗВУК

Вы уже несколько знакомы с генерацией звука на Спектруме. Самый простой пример: оператор ВЕЕР в БЕЙСИКе. С остальными звуковыми эффектами Вы знакомы лишь понаслышке из игровых программ. Что представляет из себя звук на компьютере? На рис. 1 представлено классическое изображение гармонической частоты.

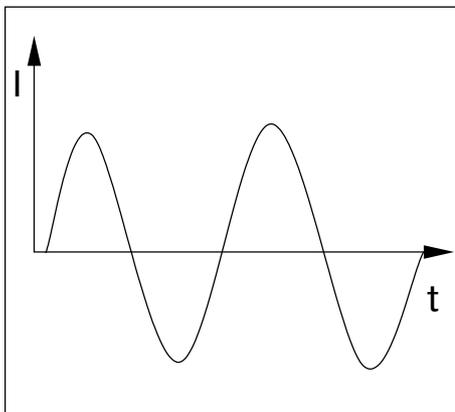


Рис. 1

На компьютере такая синусоида неформируема. Мы не сможем на "Спектруме" сформировать плавное изменение положения диффузора динамика, т.к. доступ к динамике возможен только через бит 4 (здесь четвертый бит является пятым по порядку или числом 16, если все остальные считать нулями) порта номер 254.

Положением динамика управляет один бит, значит возможны только два положения, где 0 - начальное положение, 1 - максимальное удаленное от начального положения. Графиком частоты, образованной таким образом, служит рис. 2.

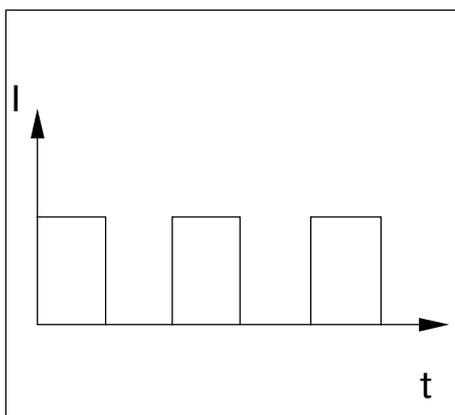


Рис. 2

Подобным образом информация считывается и записывается на магнитофон. Попробуйте набрать программу 1, в которой данные, считываемые с магнитофона, запоминаются в памяти. Информация может представлять собой человеческую речь, музыку и прочую звуковую информацию. Программа 2 проигрывает информацию, считанную с магнитофона и записанную в память, выводя ее на динамик компьютера.

В данном случае информация записывается в верхние две трети экрана с задержкой 15. Чем меньше эта задержка, тем быстрее заполнится память и более четко воспроизведение. Так что подбирайте ее по вкусу.

Программа 1.

```
TAPBUF    LD    HL, 16384
          LD    DE, 4096
SBUF      LD    C, 1
LDBYT     LD    A, 15
DELAY1    DEC   A
          JR   NZ, DELAY1
          IN   A, (254)
          RLCA
          RLCA
          RL   C
          JR   NC, LDBYT
          LD   (HL), C
          INC  HL
          DEC  DE
          LD   A, D
          OR   E
          JP   NZ, SBUF
          RET
```

Программа 2.

```
Bufdin    LD    HL, 16384
          LD    DE, 4096
STBYT     LD    A, (HL)
          LD    B, A
          LD    C, 128
PLAY      LD    A, 15
DELAY2    DEC   A
          JR   NZ, DELAY2
          LD   A, B
          AND  C
          JP   NZ, BEEP1
          XOR  A
          OUT  (254), A
BP1Q      RRC   C
          JR   NC, PLAY
          INC  HL
          DEC  DE
          LD   A, D
          OR   E
          JR   NZ, STBYT
          RET  BEEP1
          LD   A, 16
          OUT  (254), A
          JR   BP1Q
```

Но даже при минимальном значении задержки звук не будет достаточно реалистичным. Из рис. 3 видно, что выделяются не все частоты, многие из них, находясь ниже 0, приравниваются к 0, а находящиеся выше 0, приравниваются к 1.

Над данными, которые считываются с магнитофона, можно выполнять некоторые операции, например, удаление низкой частоты, подавление шума, выявление преобладающей частоты. Выявление частот используется в цветомузыке. Набрав программу 3, Вы можете узнать, сколько импульсов (точнее перепадов с 1 на 0 или с 0 на 1) произошло за некоторый интервал времени.

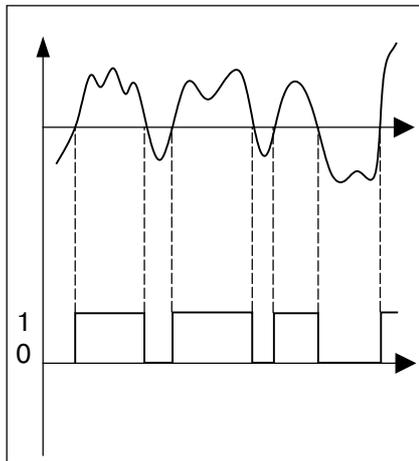


Рис. 3

Число импульсов возвращается в регистровой паре BC, т.е. Вы можете вызывать эту программу Бейсик-программой. По числу таких импульсов можно узнать, какая частота преобладает (в данном случае за 127 тактов). Чем выше частота, тем меньше значение, содержащееся в BC. Можно на основе этой программы создать цветомузыку или программу, распознающую мелодию, проигрываемую с магнитофона. Но эта программа не сможет Вам выдать точное значение частоты, идущей с магнитофона. Это значение весьма приближенно, т.к. часть частот пропадает, часть наоборот появляется в виде шума.

Программа 3.

```

BCDELAY  LD   B, 127
         LD   H, A
         LD   C, 0
IMP      18  A, (254)
         XOR  H
         AND  64
         CALL Z, INCC
         DJNZ IMPN
         RET
INCC     LD   A, H
         CPL LD
         H, A INC
         C   RET

```

Если вы, к примеру, запустив программу 1, предварительно нажмете клавишу "воспроизведение" и "пауза" на магнитофоне, то увидите, что с магнитофона, который молчит, идет шум, который весьма заметен (вполне возможно, что у Вас прекрасный магнитофон, тогда шума не будет, но урезание частот останется).

Итак, Вы представляете уже, какое качество звука можно получить, используя Спектрум, какие недостатки имеет звуковое решение на компьютере.

Но возможен резонный вопрос: "Каким же образом регулируется громкость в мелодиях игровых программ?" Что же такое громкость? С помощью рис. 4 можно увидеть, что громкость - это амплитуда звуковой частоты. Но как же можно изменять амплитуду на Спектруме?

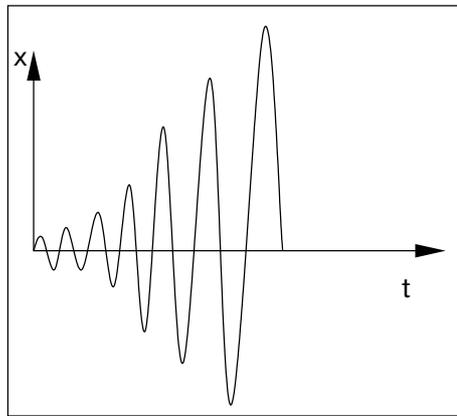


Рис. 4

Представьте себе, что на динамик подана 1. Динамик начинает поднимать диффузор. А потом тут же подадим на него 0. Диффузор опять примет начальное положение. А что, если мы будем увеличивать значение задержки между подачей 1 и 0. Диффузор поднимется уже несколько выше, и так можно увеличивать эту задержку до тех пор, пока за промежуток времени между 1 и 0 диффузор не будет успевать достичь наивысшего положения.

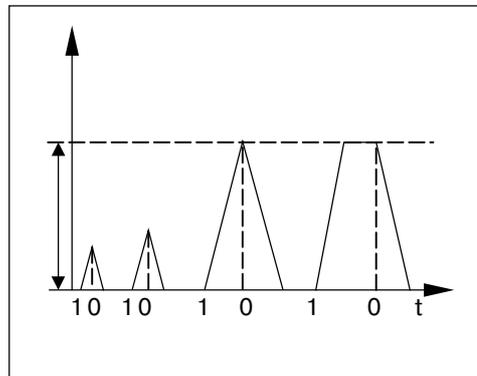


Рис. 5

Если он будет успевать, то, как такового, изменения громкости происходить не будет. Вы, наверное, уже замечали, что в играх, в которых используется затухание, слышно, что при уменьшении громкости мелодии появляется высокая частота. Это происходит именно из-за этого эффекта. Подобные мелодии с затуханиями почти невозможно слушать на маленьких (высокочастотных) динамиках, так что наиболее благородно мелодия звучит на низкочастотных (больших) динамиках. Итак, мы выяснили, что для изменения громкости нужно изменять задержку между 1 и 0.

Программа 4.

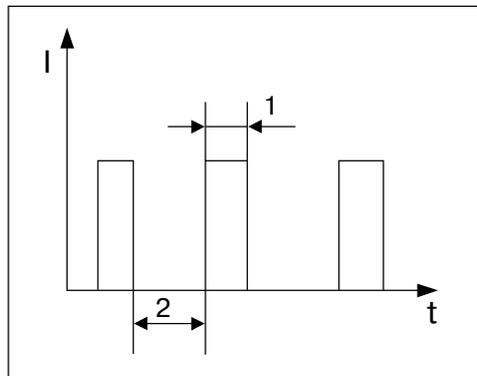
МЕТКА	АССЕМБЛЕР	КОММЕНТАРИЙ
	ORG 30000	
	DI	; Отключение прерываний.
	LD IX, 31000	; Начало данных.
CONTIN	LD A, (IX+0)	; Прием данных в аккумулятор.
	CP 255	; 255 - конец мелодии.
	JP Z, EXI	; Выход.
	OR A	; Проверка флагов в аккумуляторе.
	JR Z, N01	; Если ноль, то обход.
	LD C, A	; Временное сохранение аккумулятора.
	AND 128	; Проверка старшего (7-го) бита.
		; Если он =1, то
	LD (BARB+1), A	; включается признак ударника.
	LD A, C	; Восстановление аккумулятора.
	AND 127	; Гашение 7-го бита.
	LD D, A	; Временное сохранение аккумулятора.
	LD (NT1+1), A	; Установка звука в первом канале.
	SRL A	; Деление числа на
	SRL A	; восемь для определения

	SRL	A	; задержки громкости.
	INC	A	; Уход от возможного нуля.
	LD	(VL1+1), A	; Установили задержку громкости в ; первом канале.
	LD	A, 1	; Установка компенсационной задержки,
N01	LD	(DEL1 +1), A	; чтобы громкость не влияла на частоту.
	INC	IX	; Переход к очередному байту данных
	LD	A, (IX+0)	; (второй канал).
	OR	A	; Проверка флагов в аккумуляторе.
	JR	Z, N02	; Обход, если 0 (играть не надо).
	LD	E, A	; Временно сохранили аккумулятор.
	LD	(NT2+1), A	; Установка звука во втором канале.
	SRL	A	; Деление
	SRL	A	; на
	SRL	A	; восемь.
	INC	A	; Плюс единица (уход от нуля).
	LD	(VL2+1), A	; Установили задержку громкости во ; втором канале.
	LD	A, 1	; Установка компенсационной
N02	LD	(DEL2+1), A	; задержки.
	INC	IX	; Переход к очередному байту данных.
BEPS	LD	C, 2	; Темп
	LD	B, 0	; мелодии.
	CALL	BEEP	; Выдача сигнала на динамик.
	CALL	BEEP	; Выдача сигнала на динамик.
	CALL	DECVOL	; Уменьшение громкости.
	DEC	C	
	JP	NZ, BEPS	; 2 раза.
	JP	CONTIN	; Возврат для приема очередных данных.
; BEEP			
			; Регистр D - первый канал.
			; Регистр E - второй канал.
			; VL1 - громкость в первом канале.
			; VL2 - громкость во втором канале.
			; DEL1 - компенсационная задержка 1-го канала.
			; DEL2 - компенсационная задержка 2-го канала.
BEEP	DEC	D	
	JR	NZ, S2	
NT1	LD	D, 0	
	LD	A, 16	первый
	OUT	(254), A	
VL1	LD	A, 0	
S0	DEC	A	канал
	JR	NZ, S0	
	OUT	(254), A	
DEL1	LD	A, 32	
S1	DEC	A	
	JR	NZ, S1	
S2	DEC	E	
	JR	NZ, BARB	
NT2	LD	E, 0	
	LD	A, 16	Второй
	OUT	(254), A	
VL2	LD	A, 12	канал
S3	DEC	A	
	JR	NZ, S3	
	OUT	(254), A	
DEL2	LD	A, 32	
S4	DEC	A	
	JR	NZ, S4	
BARB	LD	A, 0	; Признак удара.
	OR	A	
	JR	Z, WAIT	
	LD	A, 16	

```

OUT    (254), A      |
LD     A, 0          |Удар
STUK   DEC          A
      JR     NZ, STUK
      LD     (BARB+1), A
WAIT   LD     A, 20
S5     DEC          A
      JR     NZ, S5
      OUT   (254), A
      LD     A, 0
      IN    A, (254)
      AND   31
      XOR   31
      JR     NZ, EXIT
      DJNZ  BEEP
EXIT   POP          BC
EXI    EI
      RET
DECVOL LD    A, (VL1+1) ; Уменьшение громкости в
      DEC   A           ; первом канале.
      JR    Z, N03      ;
      LD    (VL1+1), A ; Соответствующее увеличение
      LD    A, (DEL1+1) ; компенсационной задержки в первом
      INC   A           ; канале.
      LD    (DEL1+1), A ;
M03    LD    A, (VL2+1) ; Уменьшение громкости во
      DEC   A           ; втором канале.
      RET   Z
      LD    (VL2+1), A ; Соответствующее увеличение
      LD    A, (DEL2+1) ; компенсационной задержки во втором
      INC   A           ; канале.
      LD    (DEL2+1), A
      RET

```



1. - изменение громкости.

2. - изменение частоты.

Рис.6

Теперь перейдем к формированию частоты. Обычно звуковая частота формируется так, как показано на рис. 2. Но можно формировать и другую форму частоты. Пример на рис. 6.

Теперь уже более понятно, как формировать частоту с изменяющейся амплитудой. Если Вы сделаете программу вывода звука по этому принципу, то Ваша программа будет работать нормально. Единственный недостаток: зависимость частоты от громкости. Предположим, задержка частоты равна 10, а задержка громкости равна 20. При уменьшении громкости будет слышно только изменение частоты и лишь в конце будет слышно некоторое изменение громкости.

Перейдем к более точному методу. Чтобы частота сохранялась при изменении громкости, введем еще один параметр: задержка для сохранения частоты (b). Ее суть состоит в том, чтобы расстояние между импульсами (d) оставалось постоянным, при

изменении громкости (a). Значит, мы должны взять некоторое максимальное значение (a_{max}), откуда $b = a_{max} - a$.

Т.е. чем больше значение задержки громкости (a), тем меньше значение компенсирующей задержки (b). К примеру, для затухания (a) уменьшается на 1, а (b) увеличивается на 1 до тех пор, пока $(a) > 1$. При $(a) = 1$ затухание останавливается.

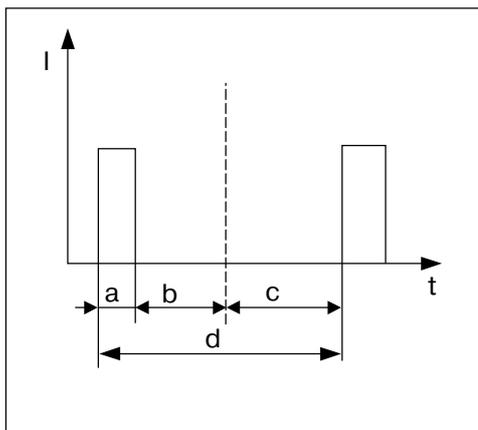


Рис. 7

Следующий вопрос: "Как сделать, чтобы одновременно могли работать несколько звуковых каналов?"

Представим себе некую сумму двух частот (рис. 8). Здесь график I - уменьшающаяся частота, график II - постоянная частота, а график S - результирующая (суммарная частота).

Каждый такой импульс представляет собой отрезок (ab) см. рис. 7. Правда, у первой и второй частоты (a) и (b) могут отличаться. Способ, предлагаемый на рис. 7 здесь не проходит. Отрезок (c) между импульсами одной частоты превращается в счетчик, который определяет: через сколько тактов должен произойти импульс. Этот способ вывода звуковой частоты реализован в программе 4.

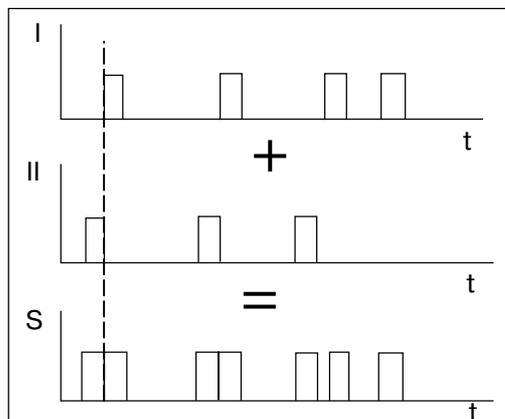


Рис. 8.

Еще одно важное замечание: отрезок (ab) занимает $1/8$ часть отрезка (c). Это необязательно принимать во внимание, но это обеспечивает нормальную регулировку громкости при более высокой частоте. А так как $256/8=32$, то задержка (ab) принята равной 32.

Как работает программа. В памяти с адреса 31000 лежит массив данных. В каждом четном байте лежит значение частоты для I канала, в нечетном - значение ноты II канала. Если номер ноты равен 0, то данные канала не иницируются. Если четный байт более 127 (установлен 7 бит), то удар в барабан. Если четный байт равен 255, то мелодия считается оконченной. Выход из программы осуществляется также нажатием на любую клавишу. Звук барабана формируется продолжительной установкой 1 на динамик и сброс затем. В конце программы подпрограмма DECVOL уменьшает значение громкости обоих каналов. Но можно сделать один из каналов с постоянной громкостью или, наоборот, с ее увеличением.

ЧИТАТЕЛЬ - ЧИТАТЕЛЮ

В силу объективных причин так сложилось, что "ИНФОРКОМ" серьезно не занимается аппаратными вопросами, связанными со "Спектрумом". Кстати, ведь каждый знает, что нельзя объять необъятное и как знать, что бы из нас получилось, если бы мы начали заниматься несвойственными нам делами, может быть и ZX-РЕВЮ уже не было бы.

Но то, что мы этим вопросом не занимаемся вовсе не означает, что мы его игнорируем. Наоборот, мы очень высоко ставим эту проблему, просто не чувствуем себя мало-мальски компетентными, чтобы самим писать на эту тему или давать какие-то рекомендации. И чем меньше мы в этих вопросах разбираемся, тем ценнее для нас каждое письмо, приходящее от наших читателей на эту тему. Особенно, если в ней затрагиваются глубинные проблемы стандартизации и обеспечения совместимости разных машин.

Сегодня своими мыслями и пожеланиями с Вами делится наш корреспондент из г. Казани Дмитрий Киселев. Это далеко не окончательная точка в проблеме. Это приглашение к свободному диалогу, к обмену идеями и мыслями, а на первых порах и это тоже большое дело. И мы полагаем, что даже тем читателям, которые и не помышляют о доработках и совершенствовании своих машин, эта статья будет интересна, поскольку они смогут узнать больше о том, в какой среде работают их коллеги, каково положение дел на сегодняшний день.

(С) Киселев Д., Казань, 1993г.

К ВОПРОСУ О СТАНДАРТИЗАЦИИ

В связи с созданием Ассоциации Пользователей Домашних компьютеров и Фонда "ФЛОП" представляю Вашему вниманию мои соображения по поводу стандартизации дальнейших расширений к ПК "Спектрум". Не секрет, что многие системные программисты уже сейчас недовольны скоростными и информативными качествами этой в общем-то неплохой машины. Пример тому - создание многочисленных режимов, не существующих в стандартном "Спектруме 48К" и даже в "128+2".

В некоторых моделях, таких как "Балтика", "АТМ" и "Пентагон профи" даже были попытки внедрить ОС CP/M (ZX-РЕВЮ 5,6-93 и "Радиоловитель N4 за 1993 г.). Однако и здесь наблюдается неразбериха в адресном пространстве этих режимов, от внешнего порта для пользователя (программатор и т.д.) на ВВ55 до разных режимов графики и цвета. Даже на ПК "ОРИОН-128", который недавно начал развиваться, уже существует стандартная таблица портов расширения ("Радио", N4, 1993 г.) Настала пора подумать о подобной таблице и для ПК типа "Спектрум". Я понимаю, что "ИНФОРКОМ" аппаратными вопросами не занимается, однако хотелось бы узнать о программных решениях этого вопроса. В частности, было бы интересно узнать о карте памяти и портах в таких ПЭВМ как "Таймекс 48 и 68" "Энтерпрайз", "Амстрад 464" и им подобных (см. Примечание "Инфоркома" в конце статьи). На Западе ведь тоже люди не только целыми днями в "Элиту" играют, но и пытаются иногда сменить картриджи и приблизиться к более серьезной среде программирования. Неплохо было бы также услышать мнение специалистов по поводу удобства программирования и логической законченности подобных расширений. Само собой разумеется, что в одном письме всех аспектов этой проблемы не отразишь, поэтому я попросил бы вкратце ознакомить читателей с этим вопросом и пусть они выскажутся и предложат свои решения. Со своей стороны я предлагаю в качестве затравки стандартизацию следующих расширений.

Во-первых, расширение памяти свыше 128К удобно адресовать через 2 старших бита в системном порту SP128 (ZX-РЕВЮ 2/91). Таким образом можно адресовать до $2^5 \cdot 16K = 512K$ ОЗУ. Однако здесь встают по крайней мере 2 проблемы:

1) Необходимо так пронумеровать страницы, чтобы пользователь мог наращивать ОЗУ постепенно: сначала 1 линейка РУ5 (я предлагаю страницы 5,2,0 закрепить за ней, а в

качестве первой сделать страницу 4, но об этом ниже). Далее при установке 2 линейки РУ5 адресовать к ней остальные 4 страницы, а при установке во 2-ую линейку РУ7 изменять только банки этих 4 страниц.

2) Так как в режиме 320К будут существовать 4 страницы с номером 7, необходима договоренность об использовании только одного SCREEN1 (ZX-РЕВЮ 2/91) или нескольких, что потребует дополнительных условий для переключения. В любом случае следует достичь максимальной совместимости с режимом 128К, а также возможности легкой доработки копировщиков, текстовых редакторов и т.д. на режим 320К. Все, кто может предложить какое-нибудь решение, пусть напишет о нем в "ИНФОРКОМ".

Теперь про теневую страницу. Номер 4 я выбрал потому, что по физическим адресам она является предыдущей к 5-й и нижние 32К переадресовывать не нужно. В теневой странице можно размещать доп. экраны, отладчики и даже альтернативные версии ПЗУ, будь то Бейсик-система с 3-х скоростной загрузкой-выгрузкой, которой я пользуюсь, или TR-DOS. Эта возможность прекрасно реализована в ПК "Балтика". Порт, расположенный по 7ЕН может не только отключать ПЗУ, но и выбирать 4 разновидности экранного пространства, о чем будет рассказано ниже.

К сожалению, при включенном ПЗУ отключается всякое обращение к этой странице, что затрудняет загрузку в нее. С помощью небольшой доработки (нужно от элемента D32 (ЛЛ1), отключить выход D33.2(ЛЕ4 выв. 6) и подключить выход D33.3 (ЛЕ4 выв. 8)) можно загружать любые данные с адреса 0000Н с помощью стандартного загрузчика в ПЗУ. Способность записывать данные в ОЗУ через ПЗУ пригодится также и для мониторов и отладчиков, имеющих собственный экран в этой области.

Кстати об экранах. В том же ПК "Балтика" реализовано 4 экрана, о чем говорилось выше. Первый экран - стандартный, расположен с адреса 4000Н. Второй экран - такой же по формату, но расположен с адреса 2000Н и может использоваться отдельным отладчиком, после этого экрана остается 2К на матрицы шрифтов 16x16. Третий и четвертый экраны заслуживают особого описания. Дело в том, что при формировании адресов их атрибутов отключается частота, делящая экран на 24 знакоместа в столбце и атрибуты адресуются так же, как и графика.

Таким простым фокусом, который легко сделать и на "Ленинграде" и на "Пентагоне 48", достигается графика не только высокого графического, но и цветного изображения (один байт пикселей на один байт цвета). Третий экран расположен с 0000Н, а четвертый с 4000Н. эти 4 экрана коммутируются битами 0 и 1, а ПЗУ отключается битом 7 порта 7ЕН. Таким образом, еще остается место для других расширений.

Я предлагаю в качестве стандартного расширения экран с 2000Н и экран высокого цветного разрешения с 4000Н, т.к. он позволяет загружать стандартные картинки, а атрибуты в крайнем случае можно или пересчитывать, или убирать совсем. Конечно, при этом будет затрагиваться область буфера принтера, но в этих областях "сорят" и многие фирменные программы. Наличие же загружаемого "ПЗУ" позволит сместить эти области вверх и полностью решить проблему.

Конечно, найдутся люди, которые скажут: - "зачем нам эти 12К графики, у нас и с 6К Спектрум не успевает работать". Однако, если стандартизировать также и турбо-режим процессора, который кстати можно прикрепить к одному из свободных битов порта 7ЕН, то соотношение 12К/7(8) Mhz будет даже лучше стандартного, да и проблема с "клэшингом" атрибутов почти решится. Остается, правда, вопрос о том, хватит ли быстродействия у ОЗУ, но это уже проблема технологов, а не программистов.

Следующий вопрос, который подымался на страницах "ZX-РЕВЮ" - это вопрос стандартизации русского шрифта. И если создать доп. матрицы знаков и разместить их в удобном месте - это не проблема, то привязку русских букв к латинским клавишам, а также вопрос о выборе способа и кода переключения нельзя оставлять без внимания. Здесь я тоже предлагаю не изобретать велосипед, а пойти навстречу рядовым пользователям - секретарям, писателям и машинисткам. Достаточно взглянуть на решение этой проблемы

со стороны фирмы IBM, которая выпускает клавиатуру именно в стандарте QWERTY (ЙЦУКЕН), а не в стандарте QWERTY (ЯВЕРТЫ), который нравится лишь программистам-системщикам, а не машинисткам. Что же касается недостающих клавиш, то здесь можно предложить ввести дополнительные линии данных в порту клавиатуры FEN. Это легко аппаратно реализуемо и используется в таких ПК, как "Оризон", "Оризон-микро", "Магик" (изг. в г. Ижевске), "Байт", а также, вероятно, в фирменном SP128+2, где есть разъем КЕУРАД. Нужно лишь стандартизировать определенную привязку этих клавиш к линиям адресов и внести небольшие изменения в ПЗУ, связанные также с кодами переключения РУС/LAT, в качестве которых я предлагаю использовать коды 02/03, которые в Спектруме не используются, а в ASCII именуется как начало и конец текста соответственно.

Необходимо также подправить синтаксис БЕЙСИКа, чтобы он не интерпретировал коды переключения внутри чисел и т.п. Впрочем, мне будет интересно узнать и другие мнения по этому поводу.

Следующая доработка не встречалась мною ни в одном компьютере и потому я рискну назвать ее своей собственной. Речь идет о расширении музыкальных возможностей Спектрума. Сегодня существует множество программ, использующих музыкальный процессор 8912(10) в режиме 48К. Может быть, они используют также и порт динамика и записи на магнитофон для имитации ударников, хотя я лично таких программ не слышал. Во всяком случае, я предлагаю использовать этот порт для создания ритм-фона и обращаю ваше внимание на следующий факт: в 8912(10) существуют 1(2) порта, которые предназначены для обмена данными с внешними устройствами и находятся внутри адресного пространства регистров 8912(10) и которые не используются.

Я предлагаю поставить на порт А (который есть на обоих процессорах) два 4-х разрядных ЦАПа, которые будут делить сигналы динамика и магнитофона на 16 уровней, что дополнительно расширит диапазон музыкальных возможностей ПК. Конечно, ЦАП это слишком громко сказано, я имею в виду всего лишь известную R-2R цепочку. Кстати, 16 уровней выходного сигнала вполне достаточно (в крайнем случае можно сделать эти 16 уровней промежуточными между "0" и "1", т.е. получим "0" плюс еще 16 уровней от +5в лог. 1), т.к. 8912(10) тоже имеет 16 уровней громкости и модуляции, во что я поверил только тогда, когда прочел его техническое описание.

Из написанного видно, что желательно проводить комплексную стандартизацию расширений для наиболее полного их использования. Теперь слово за остальными читателями. Вообще нужно активнее работать в этом направлении, чтобы потом не было мучительно больно за то, что наш любимые компьютеры - это прыжок в будущее из позапрошлого. А основными критериями для стандартизации я предлагаю сделать в порядке важности

- 1) минимальную несовместимость на программном уровне;
- 2) удобство программирования, близость в адресном пространстве однотипных устройств и расположение "опасных" портов по четным адресам, меньшим 128;
- 3) легкость аппаратной доработки на наиболее распространенных моделях.

Внедрение этих доработок позволив всколыхнуть необъятные программирующие умы наших Родин, которые наделают на них (компьютерах) такие крутые программы, что на Западе (я надеюсь) люди начнут потихоньку собирать в гаражах "СНГовские Спектрумы".

* * *

КОММЕНТАРИЙ "ИНФОРКОМа".

Читатель поставил вопрос о том, что ему интересен подход зарубежных производителей ПК "Энтерпрайз", "Амстрад-464", "Таймекс-2048" и "Таймекс-2068". Это уже вопрос по нашей части, кое-какой информацией мы располагаем.

Сразу должны сказать, что опыт "Энтерпрайза", и "Амстрада" нам мало пригодится, поскольку эти машины по архитектуре существенно отличаются от "Спектрума", несмотря

на то, что собраны на том же процессоре Z-80. То же можно сказать и о "Шарпе" и о стандарте MSX ("Ямаха", "Хит-Бит", "Спектравидео" и пр.). Все эти машины (кроме м.б. "Шарпа") уже при своем "зачатии" планировались под работу в дисковой системе, чего нельзя сказать о "Спектруме", в котором ничего для этого не было сделано, и потому это скорее промежуточные прототипы для более поздних действительно интересных (по своему) машин, и их архитектура сейчас представляет скорее академический, чем практический интерес для наших условий.

Совсем другое дело - "Таймекс". Эта фирма долгое время (до 1985 г.) работала по прямому заказу К.Синклера. Основная цель, стоявшая перед этими машинами была такая - "Обеспечить проникновение "Спектрума" на американский континент" (было открыто представительство Синклера в Бостоне). Все-таки существовала определенная разница между Европейским и Американским рынками. У американцев и требования к машинам покрупнее, да и звонкой монеты в карманах всегда было побольше. То, что эта цель достигнута не была, другой вопрос, а вот как это делалось, нам действительно может быть интересно, ведь мы тоже сейчас находимся в аналогичной позиции - обеспечиваем захват "Спектрумом" рынка СНГ.

Модель 2048 - почти полностью идентична стандартному "Спектруму". Естественно, для американцев сделали красивый дизайн, дешевая, надежная и очень удобная клавиатура и сейчас может служить отличным образцом для большинства наших производителей. Встроили порт джойстика и даже ВЫКЛЮЧАТЕЛЬ ПИТАНИЯ! Но это нам не так важно, важнее скорее то, что был применен новый контроллер экрана - на нестандартной, специально для этого случая сделанной БИС - единственное значительное изменение в архитектуре.

Благодаря ей, стало возможным (теоретически) введение дополнительных экранных режимов и второго экрана, что делалось управлением по 255-ому порту. Практически же, надо сказать, это нигде не использовалось.

Переключение экранов выполнялось командой OUT 255,1.

Кроме того, был введен дополнительный экранный режим (54 знака на 22 вместо 32X22), что исполнялось командой OUT 255,n, где n - число, которое определяет цвет символов и фона, например, OUT 255,62 - печать черным по белому.

Появилась возможность реализации цветной графики высокого разрешения. На каждое знакоместо отводилось по 8 байтов атрибутов (вместо 1), т.е. в каждой линии знакоместа можно было задавать свой цвет INK и свой цвет PAPER. Эта возможность исполнялась командой OUT 255,2.

Команда OUT 255,0 - возврат к стандартному "Спектрумовскому" экрану.

Никакой поддержки в ПЗУ для этих нестандартных режимов работы не было и они могли реализовываться только программированием в машинных кодах. Стандартное программное обеспечение, поддерживающее эти режимы также неизвестно.

Существенным шагом по изменению архитектуры компьютера была модель 2068. Она имела два встроенных порта джойстика, звуковой процессор и тот же оригинальный видеоконтроллер. Существенные изменения имело и базовое ПЗУ. Здесь действительно появилась новая карта памяти, новые системные переменные и новые порты ввода/вывода.

БЕЙСИК приобрел несколько новых команд: DELETE, FREE, STICK, ON ERROR и SOUND.

Команда DELETE позволяла удалять строку по ее номеру или группу строк (DELETE.....).

Команда FREE позволяла узнать объем свободной памяти БЕЙСИКА, например (PRINT FREE).

Команда STICK читала положение одного из двух джойстиков и работала аналогично команде INKEY\$. Ее формат, например:

PRINT STICK (m,n)

Здесь $m=1$, если интересуется положение рукоятки или $m=2$, если интересуется состояние кнопки "ОГОНЬ". Параметр $n=1$ или 2 (соответственно левый или правый джойстик).

Команда ON ERROR обеспечивала переход в заданное место программы в случае возникновения ошибки. Команда SOUND служила для управления звуковым процессором, который имел три музыкальных и три шумовых канала, из БЕЙСИКа.

Как и в "Таймексе-2048" здесь возможны те же режимы управления экраном, тоже возможно иметь два экрана или высокое разрешение атрибутов или 64 символа в экранной строке.

Соответственно, была изменена карта памяти (звездочками отмечены участки, совпадающие со стандартным "Спектрумом").

Карта памяти с одним экраном.

0000H	Системное ПЗУ. (*)
1000H	Графический экран. (*)
5800H	Атрибуты. (*)
5B00H	Буфер принтера. (*)
5C00H	Системные переменные.
6000H	Машинный стек.
6200H	Диспетчер прикладных программ в машинных кодах.
6315H	Код управления банками памяти.
6840H	Блок переменных для прикладных программ в машинных кодах.
ARSBUF	
CHANS	Информация о каналах.
PROG	БЕЙСИК - программа.
VARs	Переменные БЕЙСИКа.
ELINE	Буфер редактируемой строки.
WORKSP	Буфер INPUT.
STKBOT	Стек калькулятора.
STKEND	Свободно.
RAMTOP	Верхний предел БЕЙСИКа.
UDG	Графика пользователя.
P_RAMT	Физический конец ОЗУ.

Как видно из этой карты памяти, во многом ее структура совпадает с картой стандартного "Спектрума". В то же время, выделение зарезервированных областей для машинного стека, блока диспетчера прикладных программ в машинных кодах и блока машиннокодовых переменных, говорит о том, что во-первых, была попытка стандартизировать программы пользователя в кодах и, во-вторых, по-видимому, предполагалось в дальнейшем надстраивать архитектуру компьютера, оставив эти области

неизменными для обеспечения совместимости с ранее выпущенным программным обеспечением.

Переключение графических режимов выполнялось теми же командами OUT, что и на Таймексе-2048, но здесь это поддержано новой картой памяти и новыми системными переменными.

Карта памяти с двумя экранами.

0000H	Системное ПЗУ. (*)
4000H	графический экран-1. (*)
5800H	Атрибуты-1. (*)
5B00H	Буфер принтера. (*)
5C00H	Системные переменные.
6C00H	Графический экран-2. Атрибуты - 2.
7B00H	Блок переменных для прикладных программ в машинных кодах.
ARSBUF	
CHANS	Информация о каналах.
PROG	БЕЙСИК-программа.
VAR5	Переменные БЕЙСИКа.
ELINE	Буфер редактируемой строки.
WORKSP	Буфер INPUT.
STKBOT	Стек калькулятора.
STKEND	Свободно.
RAMTOP	Верхний предел БЕЙСИКа.
UDG	Графика пользователя.
F7C0H	Машинный стек.
F9C0H	Диспетчер прикладных программ в машинных кодах.
FAD5H	Код управления байтами памяти.
P_RAMT	Физический конец ОЗУ.

Новые порты I/O.

В Таймексе-2068 было задействовано несколько новых портов ввода/вывода для обслуживания во-первых двух джойстиков и, во-вторых, для обслуживания музыкального процессора.

Управление музыкальным процессором осуществлялось командами OUT по портам F6H - регистр данных и F5H - индексный регистр. Сначала командой OUT по порту F5H получим доступ к одному из регистров музыкального процессора, а потом командой OUT по

порту F6H -вводим в него нужное число.

Для чтения джойстиков сначала включался контроллер командой
OUT F5H,DEH

а затем их положение определялось анализом портов:

F5FEH - левый

F6FDH - правый.

Дополнительные возможности.

Из-за существенных изменений в ПЗУ, в системных переменных и в карте памяти стала проблематичной совместимость с программами, написанными для фирменного "Спектрума". В принципе, эта совместимость возможна, но это скорее исключение, чем правило. Для того, чтобы все-таки добиться полной совместимости, фирма "Таймекс" начала выпускать дополнительные картриджи ПЗУ, для подключения которых компьютер имеет специальный отсек.

После подключения такого картриджа компьютер становится полностью аналогичен фирменному варианту, сохраняя свои дополнительные возможности для тех, кто обращается к ним через внешние порты из машинного кода.

Для удобства пользователей на подобных картриджах выпускались и игровые и прикладные программы, например текстовый редактор, а с учетом того, что американцы испытывали в то время явную ностальгию по CP/M, был выпущен и картридж "Зебра", превращающий компьютер в CP/M-совместимую машину.

У наших программистов, имеющих со старых времен большие заделы работ в CP/M тоже явно прослеживается ностальгия по этой системе, и вообще во многом пути развития могут быть схожи. Сейчас не хотелось бы делать выводы и рекомендации, еще не время, но опыт предшественников может дать пищу для размышлений...

Единственное, что можно было бы сказать тем, кто думает над доработками и стандартизацией. Опыт показывает, что это не должно быть самоцелью и обязательно должно быть **ПОДДЕРЖАНО ЭКОНОМИЧЕСКИ**.

Пусть мы скажем грубо, но прямо. Вы можете быть альтруистом сколько хотите и можете делать какие угодно открытия просто за спасибо, из любви к работе. Но если то, что Вы придумали, не даст другим людям заработать ни копейки, Ваше дело не приживется. И наоборот, если Вы открываете новый путь, по которому пойдут люди и увидят для себя в этом возможность не только поработать, но и **ЗАРАБОТАТЬ**, они подхватят, разовьют и увековечат Вашу идею и Вас вместе с ней. На этом построен весь многолетний опыт К. Синклера.

Он дал возможность развернуться тысячам фирм, выпускающим программы и сотням фирм, выпускающим периферию. Те сделали свой капитал, а его компьютер стал самым популярным в мире. То же самое происходит и у нас.

Впрочем, последнее слово останется конечно за Вами, дорогие читатели и мы будем ждать Ваши письма и Ваше мнение.

От редакции.

Представляя читателям эту интересную авторскую работу, мы должны сказать несколько слов о "Спектруме", о нашем взгляде на этот компьютер и вообще на "Синклеровское" движение в целом.

"Спектрум" занимает совершенно особое место в деле компьютеризации нашей страны. Количество его поклонников давно перевалило за миллионы. Это движение, если говорить честно, стало всенародным и уникальным. Уникальность его состоит в том, что огромная часть самого талантливого, способного и одаренного населения одной шестой земного шара, не спрашиваясь ни советов, ни разрешений у высокого начальства, сделала то, что оказалось не под силу никаким госкомитетам и институтам. В рамках скромных 48 килобайт каждый день во всех уголках страны решаются интересные проблемы, делаются настоящие открытия.

Отличительной чертой нашего "синклериста" стала необычайная гордость за свою машину и уважение к ее возможностям. Ни для кого не секрет, что это маленькое чудо может очень многое из того, что и предположить себе не могут фирмы, торгующие суперсовременными 386-ыми и 486-ми IBM-совместимыми машинами. Мы знаем множество примеров, когда способные программисты, намаявшись за рабочий день с IBM-совместимой техникой, приходят домой и отдыхают наедине с любимым "Спектрумом". Они ни на что его не променяют.

Умение реализовать свои идеи, найти новые решения, сделать свой компьютер профессиональным, оставаясь в пределах 48 килобайт, дают такой большой простор для творчества и самоутверждения, какого не во всяком деле найдешь. Мы можем восхищаться гигантскими прокатными станами, ловко расправляющимися с тоннами металла, но всегда будет в почете профессия ювелира, способного так обработать несколько граммов, что они будут радовать глаз и вызывать всеобщее восхищение.

Все это имеет непосредственное отношение к данной статье. Астрономические расчеты достаточно сложны, но это одна из тех областей, в которых Ваш "Спектрум" смело может потягаться с гораздо более мощными потомками. Редакцию "ZX-РЕВЮ" особенно радуют такие материалы, которые выводят эту машинку на новый уровень, позволяют взять от него то, для чего она не предназначалась.

Мы никогда не сталкивались с астрологией и были приятно обрадованы тем фактом, что это, как оказалось, не мертвая наука былых лет, а постоянно развивающееся учение и занимаются им люди серьезные. Они непрерывно растут, повышают свою эрудицию, компьютерную грамотность, расширяют круг общения. Короче говоря, если Вы растете и вместе с вами растут возможности Вашего "Спектрума", то это то, что надо - основная цель "ZX-РЕВЮ" достигнута!

Для тех же из Вас, кто захочет заняться этим делом основательно, эта статья сможет стать первой вешкой на длинном и трудной пути. Мы далеки от мысли, что прочитав и разобрав ее, Вы сразу выйдете на некоторый достаточный уровень. Нет, это будет только началом. Может быть, даже трудным началом. Вам еще предстоит узнать и понять очень и очень многое. Статья достаточно сложна для неподготовленного читателя, поэтому кое-что, самое необходимое, в процессе редактирования мы вынесли в отдельный комментарий в ее заключении.

Автор статьи - Сергей Крушинский. Он не является ни профессиональным астрологом, ни профессиональным программистом. И то и другое - хобби. По профессии он журналист. В то же время, его программы по астрологии у нас распространены не меньше, чем западные аналоги и, если Сергей достиг значительных успехов в своем деле, то благодаря огромной любви к астрологии и, самое главное, к своему SPECCY, как он нежно называет "Спектрум".

SPECCY + АСТРОЛОГИЯ



Доводилось ли когда-нибудь читателю заглядывать в кабинет современного астролога? Если да, то могу спорить: первое, что бросилось в глаза, было не магическим кристаллом и не чучелом крокодила. На самом почетном месте светился дисплей! Как-то у моего знакомого, считающегося по праву одним из ведущих московских астрологов, сломался его "Коммодор". "Такое чувство, словно выключилось левое полушарие моего мозга", - жаловался он. Ничего удивительного, я сам вспоминаю как кошмар возню с таблицами и циркулем, длившуюся порой до полуночи ради того, чтобы построить единственный гороскоп, а под утро, напрягая взгляд, пытаешься его прочитать.

Дело решительным образом изменилось после того, как я купил "Спектрум" и написал первую программу - она стала делать за меня "грубую" работу.

Что такое компьютерная астрология? Это эффективная работа с большим количеством информации, возможность экспериментировать и исследовать. Чего только нет в современных астрологических программах, написанных для машин класса IBM: базы данных, атласы мира, графика, статистическая обработка информации... Даже самому отъявленному скептику небезынтересно взглянуть на аккуратные распечатки, представляющие его персону с новой точки зрения.

Может показаться, что "Спектруму" с его 48-ью килобайтами не под силу тягаться с куда более мощными компьютерами. Ничего подобного. Даже если за дело возьмется начинающий программист, не имеющий понятия ни о чем, кроме стандартного Бейсика - каким я был 2 года назад, когда написал первую программу, - SPECCY хватит на то, чтобы сделать по крайней мере самое важное, а именно: построить гороскоп рождения.

Что такое гороскоп.

Астрология покоится на аксиоме, согласно которой состояние мира в данном месте и в данный момент времени есть нечто уникальное. Когда человек рождается, он настолько восприимчив, что это состояние мира накладывает на него неповторимый отпечаток, впоследствии в большей или меньшей степени определяющий его реакции, а также тип ситуаций, с которыми приходится сталкиваться.

Самая большая роль отводится Солнцу, Луне и планетам Солнечной системы. Их расстановка - основа астрологического "сюжета" - никогда не повторяется во всех подробностях, так же, как не повторяется ни одна шахматная партия.

Сравните три рисунка. На них изображено по сути одно и то же: так называемая "небесная сфера". Вторая картинка имеется в любом астрономическом учебнике. За ее разъяснением можете обратиться и в энциклопедию. А третья - традиционный гороскоп.

Положение любого тела на небесной сфере определяется парой координат: долготой или угловым расстоянием от точки весеннего равноденствия до проекции тела на эклиптику (* / 1), и широтой - расстоянием "вверх" или "вниз" от нее, со знаком "+" к Северному полюсу эклиптики и "-" к южному.

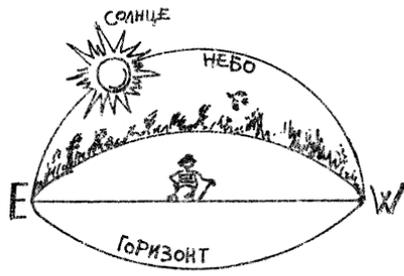


Рис. 1

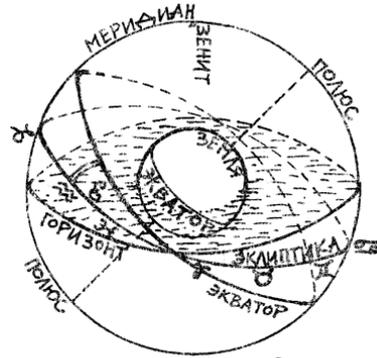
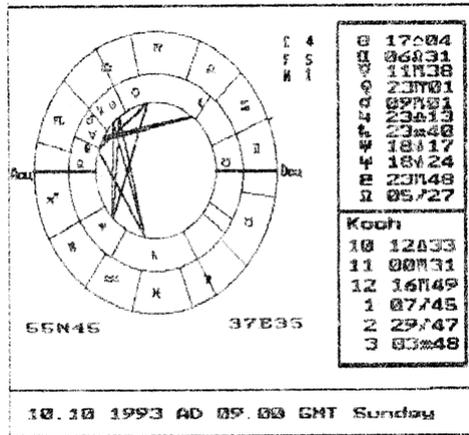


Рис. 2



Внешний круг - это пояс Зодиака. Он делится на 12 знаков:

- | | |
|--------------|--------------|
| ♈ - Овен | ♉ - Телец |
| ♊ - Близнецы | ♋ - Рак |
| ♌ - Лев | ♍ - Дева |
| ♎ - Весы | ♏ - Скорпион |
| ♏ - Стрелец | ♐ - Козерог |
| ♑ - Водолей | ♒ - Рыбы |

Внутреннее кольцо разделено на 12 домов. Их тоже 12. Их отсчет начинается с линии горизонта (Асц-Дсц) и нумеруются против часовой стрелки, как и знаки.

В этом же кольце расположены Солнце ☉, Луна ☾ и планеты:

- | | | | |
|----------|---|--------|---|
| Меркурий | ☿ | Венера | ♀ |
| Марс | ♂ | Юпитер | ♃ |
| Сатурн | ♁ | Уран | ♅ |
| Нептун | ♆ | Плутон | ♇ |

Кроме того, Северный или Восходящий Лунный узел ♁.

Зодиак - это пояс созвездий, тянущийся вдоль эклиптики. По нему совершают свой видимый путь Солнце, Луна и планеты. Астрологи делят его на 12 знаков по 30 градусов: Овен, Телец, Близнецы и т.д. Начало Овна - точка весеннего равноденствия. Астрологи говорят: "Марс - 12 градусов Тельца", а не "42 градуса", хотя это то же самое. В наше время знаки не совпадают с одноименными созвездиями.

Другая система координат называется экваториальной. В ней две мерки - прямое восхождение и склонение.

В астрологии есть еще такое понятие, как "дома". Их, как и знаков, двенадцать, только получают они путем деления не зодиака, а пространства вокруг горизонта. Начало 1-го дома совпадает с точкой горизонта, где восходит солнце. Если знак говорит о том, как проявляет себя находящаяся в нем планета, то дом - о сфере жизни, на которую это действует. Сами же планеты - символы определенных жизненных принципов.

Солнце - источник сил, достоинства, принцип власти: глава государства и "царь в голове" (этот ряд можно продолжать до бесконечности); Луна - инстинкты, привычки, чувства, Марс - борьба, самоутверждение, Венера - гармония, любовь и т.д. Изучив положение планет в знаках, домах и угловые расстояния между планетами, астролог пытается построить характерный сюжет.

... Итак, надо научить "Спектрум" делать следующее:

1) Рассчитывать эклиптические геоцентрические (* / 2) координаты Солнца, Луны и планет;

2) Определять границы домов в той же эклиптической системе координат и выводить результат на экран или принтер хотя бы в виде таблицы. Впрочем, места вполне хватит и на приличный рисунок.

Основой для расчетов будут служить дата, время рождения и, кроме того, географическая широта и долгота. Рассмотрим подробнее эти этапы.

Основной расчет.

Вначале определяется положение Солнца на эклиптике. Рассчитываем ZM (среднюю аномалию), ZV (истинную аномалию), ZR (радиус-вектор). Затем довольно простым способом SL, долготу солнца. Широта его всегда равна нулю (* / 3). Две величины, полученные на данном этапе, ZR и ZM надо сохранить как SR и SM - они понадобятся в дальнейшем. Чтобы найти положение планеты, необходимо:

1) Рассчитать тем же способом, что и для Солнца, ZM, ZV, ZR;

2) Найти гелиоцентрическую широту (* / 4) и долготу планеты (ZB и ZL);

3) Используя полученные ранее величины, перейти к геоцентрическим координатам.

Местонахождение Луны определяется совершенно особым способом, для нее придется предусмотреть отдельный программный блок. Вместе с Луной будет рассчитываться и долгота лунного узла - точки, в которой Луна пересекает эклиптику, двигаясь с юга на север. Она важна для гороскопа, поскольку говорит о том, насколько гармонично личность вписывается в социум. Все сказанное можно проиллюстрировать схемой (табл. 1):

Таблица 1

Солнце	Планеты (от J=2 до J=9)	Луна
SL, SR, SM A(1,1)=SL A(1,2)=0	ZL, ZB A(J,1), A(J,2)	A(10,1), A(10,2) A(11,1) A(11,2)=0
ZM, ZR, ZV		

Входом в этот блок процедур послужит переменная, выражающая искомый момент времени - T и таблица, где содержатся характеристики орбит на некоторый известный момент, так называемую начальную эпоху T0. Их нетрудно пересчитать на эпоху T. Выход - массив A(2,11), долгота и широта 10 небесных тел и лунного узла.

Формулы, которые используются для решения почти всех поставленных здесь задач, - это классические формулы небесной механики, основывающиеся на законах, открытых Иоганном Кеплером. Смешно подумать, они умещаются в десяток Бейсик-строк. Есть, правда, нюанс. Тела Солнечной системы влияют друг на друга, что нарушает в некоторой мере "правильность" их движения. В астрономии это называется "возмущениями". Сатурн, например, может оказаться в соседнем градусе. Для достижения точности, на разных этапах расчетов приходится вносить поправки.

Чтобы вычислить положения планет с точностью до угловой секунды, рекомендую

такой алгоритм:

- а) отключить телефон, запереть дверь;
- б) выпить крепкого кофе;
- в) набирать тригонометрические выражения, страницу за страницей; вернуться к шагу б).

Выход из этого цикла надо предусмотреть по одному из следующих условий:

- если не хватило памяти компьютера;
- если у программиста нервное истощение;
- если последний убедился, что его старания бесполезны, т.к. все равно для 8-разрядного компьютера сверхвысокая точность недостижима (или достижима, но сверхвысокой ценой)!

К счастью, в астрологии, в отличие от навигации, не требуется такая точность. Плюс-минус 10 минут вполне достаточно. Поэтому надо будет всего лишь внести маленькую поправку в гелиоцентрическую долготу Сатурна, Юпитера, а по желанию и Нептуна и Урана. Эта точность будет почти неизменной не только для нашего столетия, но и, скажем, для 16-го века.

Календарь.

Период T говорит о том, сколько столетий прошло от начальной эпохи. Примем за начальную эпоху 31 декабря 1889 г., 12 часов 00 минут по Гринвичу. Рассчитаем, сколько суток и долей суток прошло с этого момента. Переведем их в столетия, поделив на 36525. Если наша дата приходится на один из прошлых веков, число будет отрицательным.

Определение интервала в сутках - дело довольно хлопотное, если браться за него "в лоб". Поэтому астрономы применяют так называемую систему юлианских дат. За точку отсчета взят средний Гринвичский полдень 1 января 4713 г. до н.э. В следующий полдень наступила юлианская дата 1, потом -2 и т.д. Таким образом, начальной эпохе соответствует ю.д. 2415020, а 1 февраля 1965 г. 11 ч. 46 мин. - 2438792.99. Для определения ю.д. существует формула:

$$JD = \text{TRUNC}(365.25 * Y) + \\ \text{TRUNC}(30.6001 * (M+1)) + \\ D + A + 1720994.5$$

Если месяц - январь или февраль, то $Y = (\text{год} - 1)$, $M = (\text{месяц} + 12)$. Иначе $Y = \text{году}$, $M = \text{месяцу}$.

$$D = \text{число} + (\text{час по Гринвичу} + \text{мин} / 60) / 24$$

$$A = 0, \text{ если дата дана по старому стилю. Иначе } A = 2 - X + \text{TRUNC}(X/4), \text{ где } X = \text{TRUNC}(Y/100)$$

Дальше просто: вычитаем из ю.д. 2415020 - получается число суток от начальной эпохи. Между прочим, в подавляющем большинстве программ, занимающихся определением дня недели и биоритмов, используется система юлианских дат. Так, если известна ю.д. в 0 часов по Гринвичу, то номер дня недели:

$$N = (JD + 1.5) \text{ MOD } 7.$$

0 - воскресенье, 6 - суббота.

Примечание: Здесь мы применили две функции - $\text{TRUNC}()$ и MOD , которых нет в стандартном Бейсике, хотя они имеются среди кодов калькулятора "Спектрума", поэтому надо дать некоторые пояснения.

Функция TRUNC выделяет целую часть дробного числа. Дотошный читатель может предположить, что то же самое делает и стандартная функция INT , но это не совсем так. Они совпадают только для положительных чисел, для отрицательных же они различаются.

Дело в том, что INT всегда округляет число "вниз", а TRUNC - всегда округляет к "нулю".

Так, если число равно 3.14, то $\text{INT}(3.14) = 3$ и $\text{TRUNC}(3.14)$ тоже равно 3. С другой стороны, если число равно -3.14, то $\text{INT}(-3.14) = -4$, а $\text{TRUNC}(-3.14)$ равно -3.

Задать в стандартном БЕЙСИКе функцию TRUNC можно так:

```
DEF FN T(X)=SGN(X)*INT(ABS X)
```

Теперь функция $\text{MOD}(X,Y)$ - это остаток от деления X/Y :

```
DEF FN M(X,Y)=X-Y*INT(X/Y).
```

Итак, помимо блока с основными расчетами, в программу надо включить календарные вычисления. На входе дата и время по Гринвичу, на выходе - T (период) и JD (юлианская дата, она может еще пригодиться).

Ввод исходных данных.

Если не претендовать на очень уж товарный вид программы, то подойдет стандартное INPUT. Обозначим данные, которые должны быть получены в результате так: YE, MO, DA, HO, MI, LA, LO - год; месяц; день; часы; минуты; градусы широты (положительное число, если она северная, отрицательное, если южная); градусы долготы ("+" к западу, "-" к востоку, это может не соответствовать некоторым стандартам, но так удобнее).

Способ организации ввода, т.е. вводить ли поясное время, а потом пересчитывать его в Гринвичское или сразу последнее; записывать долготу в виде градусов/минут или в виде десятичной дроби, как вводить для нее знак - дело вкуса. Главное - разобраться во всем заранее, в противном случае появятся ошибки, которые трудно будет выловить. И, разумеется, надо своевременно сообщить пользователю о выбранном формате.

В англо-американских программах принято записывать не "23:15", а "11:15", затем "PM" - "после полудня". Потом - номер часового пояса ("TIME ZONE"). 16 июня 1930 г. почти на всей территории бывшего СССР стрелки были переведены на 1 ч. вперед, что получило название "декретное время". А с 1980-го действует летнее. Поэтому к номеру часового пояса надо еще прибавлять 1 или 2 часа.

Координаты Москвы в большинстве "фирменных" программ записываются так: 037E35; 55N45 ("N" означает север, "E" - восток).

Я вполне солидарен с программистами, которые считают хорошим тоном проверять, не ошибся ли пользователь. Широта не может выходить за пределы -90, +90 градусов. В феврале 1993 г. 28, а не 29 дней... Что еще хорошо бы учесть? В самом начале я писал о домах гороскопа. Сегодня чаще всего применяют 2 системы: Планидуса и Коха, и в том и в другом случае при расчете идет в ход тригонометрия. Так вот, если задана заполярная широта и одна из этих систем, происходит остановка с сообщением "INVALID ARGUMENT".

Определить, сколько дней в любом месяце любого года можно по формуле:

$$X=28+ABS(INT(MO/8)-(MO/2-INT(MO/2))*2)+2*SGN(ABS(MO-2))+(1-SGN((Y/4-INT(YE/4))*4))*(1-SGN(ABS(MO-2)))$$

В тех же фирменных программах, как правило, имеется банк данных, в котором хранятся во-первых имена и даты рождения клиентов, а во вторых, координаты городов. Дисковая операционная система TR-DOS позволяет сделать то же самое на "Спектруме".

Астрологические дома.

Если полный оборот вокруг своей оси Земля совершает примерно за 24 часа, то каждые 4 минуты, или около того, над горизонтом восходит новый градус эклиптики. В астрологии этот участок называется Асцендентом и считается одним из самых важных факторов гороскопа, пожалуй даже более важным, чем солнечный знак (о котором идет речь на последних страницах газет). Примерно так же меняется кульминирующий градус или середина неба (она занимает самое высокое положение в небе по сравнению с остальной частью эклиптики). Противоположные им точки называются "десцендент" и "Основание неба", а все вместе - "углами гороскопа", вот вкратце их значения:

Асцендент - темперамент, внешность, самооценка, самые характерные черты

личности и судьбы... вообще, по одному нему можно сказать очень многое о человеке. Вся астрология, если угодно - не что иное, как учение о больших и малых циклах. Асцендент - это начало (в этой точке планета восходят, "рождаются"), а по началу любого цикла можно спрогнозировать его дальнейшее развитие. Еще его можно сравнить с оглавлением книги, где коротко указано, чего от нее ждать.

Середина Неба - социальная реализация личности, карьера, общественное положение (здесь небесные тела кульминируют; Солнце пересекает это место в полдень, образно говоря, достигает расцвета), С.Н. в решительном, инициативном знаке Овна - человек готов идти на риск и борьбу, добиваясь профессионального успеха; в знаке Рыб - наоборот, пассивен, плывет по течению, зато шансы нередко сами приходят к нему в руки...

Десцендент - говорит о других людях (в противовес "Я" - Асценденту), с которыми приходится считаться. Прежде всего, это партнеры: коллеги, супруг.

Основание Неба - "источник": родительский дом, детство, родина, "семейное гнездо".

Асцендент - начало 1-го дома. О.Н. - 4-го, Десцендент - 7-го, С.Н. - 10-го. Границы остальных домов рассчитываются в разных системах по-разному. На протяжении 2 тысячелетий нашей эры одни системы уступали место другим. Сегодня широко известны несколько. Любопытно, что одна из них принадлежит перу Иоганна Молиториса де Кунигсберга (1436-1476), признанного математика, внесшего вклад в сферическую тригонометрию. Его имя, как и псевдоним - Региомонтанус - говорит о том, что родом он из нынешнего Калининграда.

Другая система приписывается т. Кампанелле... Я приведу систему Вальтера Коха (1895-1977). Считается, что она особенно хорошо работает в сфере бытовых прогнозов.

Расположение домов неодинаково не только в разное время, но и на разных широтах. Представьте себе апельсин, разрезанный наискосок.

От угла, под которым прошел нож, будет зависеть внешний вид содержимого. Помимо широты и времени суток для решения задачи надо знать еще период T и наклон эклиптики в эпоху (угол ϵ , см. рис. 2) Обозначим его EC .

$$EC = 23.452292 - 1.30125E-02 * T - 1.6388889E-06 * T * T + 5.0305556E-07, \text{ градусов.}$$

Сперва определяется прямое восхождение Небесного меридиана $RAMC$, затем прямое восхождение пересчитывается в долготу (т.е. переходим от одной системы координат к другой) - получается С.Н. Находим асцендент. Наконец, зная "углы", одну за другой рассчитываем границы домов 11-го, 12-го, 2-го и 3-го. Остальные дома расположены в противоположных градусах. На выходе - массив $H(1...12)$, дома.

"Дебют".

А.Г.Алексеев в одной из статей для "ZX-РЕВЮ" удачно назвал этим словом фрагменты, которые повторяются в разных программах - с них удобно начинать. В мой "дебют" входит несколько процедур:

1. "INTERVAL".

В астрономических вычислениях часто приходится иметь дело с большими углами. Например, угол ZM , рассчитанный по стандартной формуле, может оказаться равным 29345.672 градуса или 512.17858 радианам. Это то же самое, что 185.672 градуса или 3.2405876 радиан. Мало того, что последние два числа нагляднее. Бывают случаи, когда вычисляя тригонометрические функции больших углов, калькулятор "Спектрума" ошибается.

Попробуйте, например, PRINT SIN(512.17858).

Получится -.098815626.

Затем: PRINT SIN(3.2405876).

Результат: -.098833433! Между прочим, IBM показывает: -0,0988334620170090.

Десяток таких ошибок, и результат окажется столь далеким от точного, что духи-

управители планет разгневаются на астролога, который им воспользуется... дабы избежать их немилости, предлагаю процедуру, которая приводит число A в интервал 0 - AMAX:

```
95 REM: вход A-число, AMAX-верхний предел
96 REM: X-результат, Y-локальная переменная
100 LET Y=INT(ABS(A)/AMAX)*AMAX
110 LET X=A-Y*(A>=AMAX)+(AMAX+Y)*(A<0): REM: Слава Джорджу Булю!
120 RETURN
```

2. "POLYNOME"

Как Вы заметили, ряд формул содержит умножение последовательности чисел на степени времени (T). Процедуру для этого на Бейсике удобно оформить в виде функции пользователя:

```
129 REM POLYNOME
130 DEF FN P(A,B,C,T)=A+T*(B+C*T)
```

3. "ARCTANGENS(A/B)"

При использовании функции ATN возникает неопределенность: неизвестно, в каком квадранте лежит вычисленный угол. Его можно установить по знакам A и Y:

```
139 REM на входе A,B (радианы), на выходе X
140 IF NOT B THEN LET B=1E-6: REM на случай, если B=0
150 LET Y=ATN(A/B)
160 LET A=Y+(Y<0)*PI+(A<0)*PI
170 LET B=PI+PI: GOSUB 100: REM к процедуре INTERVAL, которая вернет результат X
180 RETURN
```

Кроме этих подпрограмм, я включаю в "дебют" объявление констант.

```
10 RESTORE 9000
20 READ ZE, ON, L6, K
30 LET S=PI/K: REM величина, приближающаяся к 1 радиану. Чтобы переводить градусы в радианы
    и наоборот, удобнее всего умножать, и делить число на S, а не на (PI/180)
40 LET K2=K+K: REM 360
50 LET P2=PI+PI
9000 DATA 0,1,60,180: REM константа
```

Этот ход значительно сокращает программу. Все же, из соображений читаемости, я не буду использовать константы в примерах.

Неплохо было бы написать процедуры "дебюта" в кодах калькулятора и разместить их, например, в REM-строках. Параметры удобно передавать туда через функции пользователя. (См. "Элементарная графика", гл.3. или "Программирование в машинных кодах и на языке АССЕМБЛЕРА", ч. 2. гл. 4.6). Вопрос в том, как возвращать в Бейсик результат, он ведь находится на стеке калькулятора в интегральном представлении. Я знаю несколько способов. Увы, ни один из них не назовешь вполне изящным. Самый примитивный - завести особую переменную, допустим Q, затем написать на Ассемблере процедуру, которая будет искать в области Бейсик-переменных нашу Q и переносить в отведенные ей 5 ячеек памяти 5 байт с калькуляторного стека. Процедура, которую я привожу - это слегка измененная программа FIND из N-3,4 "ZX-РЕВЮ" за 1993 год ("Применение ассемблера для создания быстроработающих программ").

```
CALL FIND      ;Вызов процедуры, которая ищет адрес переменной и помещает его в HL.
INC HL        ;После кода имени переменной идут 5 байтов, туда и надо переместить 5
              ;байтов со стека калькулятора.
EX DE,HL     ;Чтобы выполнить команду LDIR, этот адрес должен быть в DE.
LD HL,(5C65) ;Теперь надо найти начало стека калькулятора. Помещаем в HL
```

; содержимое системной переменной STKEND. Число находится в ячейках от
;(STKEND)-5 до (STKEND)-1.

```
DEC HL
DEC HL
DEC HL
DEC HL
DEC HL ;Теперь в HL - адрес первого байта искомого числа.
LD BC,5 ;Его длина - 5 байт.
LDIR ;Передаем его переменной.
RET
```

```
-----
FIND ;Поиск переменной.
LD HL,(5C4B) ;Адрес системной переменной VARS.
X1 LD A,51H ;Код буквы "Q"
CP(HL) ;Сравнение и выход, если переменная найдена. Ее адрес - в HL.
RET Z
BIT 5,(HL) ;Если 5-й бит в имени равен 0, то это либо массив,
;либо символьная строка.
JR NZ,X2 ;Следующие строки следует пропустить.
INC HL
LD E,(HL)
INC HL
LD D,(HL)
ADD HL,DE
INC HL
JR X1

X2 BIT 6,(HL) ;Аналогичным способом проверяем, не является ли переменная
;многосимвольной, параметром цикла, наконец, если ни тем, ни
;другим - соответствует ли ее имя "Q".

JR NZ,X3
INC HL
LD A,(HL)
BIT 7,(HL)
JR Z,-6
LD DE,6
ADD HL,DE
JR X1

X3 BIT 7,(HL)
JR Z,-10
LD DE,13
ADD HL,DE
JR X1
```

Конструкция получится такая:

1. В БЕЙСИКе: Объявить переменную Q

```
LET Q=0
```

Объявить функцию пользователя:

```
DEF FN A(A,B,...)=USR адрес
```

Вызывать ее:

```
LET X=FN A(...)
```

X здесь не имеет ровно никакого значения: он показывает лишь содержимое регистровой пары BC. Важно то, что ответ содержится в Q.

Читателям, не имеющим опыта работы с калькулятором "Спектрума", предлагаю воспользоваться отлаженной процедурой INTERVAL. Перед тем, как вызывать ее, надо поместить параметры на калькуляторный стек. Первый параметр на стеке (сверху) - АМАХ, следующий - число А (листинг 1).

При вычислении Y неплохо было бы воспользоваться командой mod/div, которой располагает калькулятор, но, к сожалению, она не во всех случаях работает корректно.

Дес. код	Мнемоника	Комментарий. Содержимое стека.
6		;Поскольку будет использоваться операция
10	LD B,0A	; ">=", надо подготовить регистр В, помес-
		; тив туда ее код (10)
239	RST 28H	; Включение калькулятора.
192	store M0	; Сохраняем АМАХ в ячейке M0
2	delete	; и удаляем его со стека.
193	store M1	; Сохраняем А в ячейке M1
43	abs	; ABS(A)
224	recall M0	; Вызов АМАХ на стек
5	divide	; ABS(A)/АМАХ
50	truncate	; TRUNC(ABS(A)/АМАХ)
224	recall M0	; АМАХ, TRUNC(ABS(A)/АМАХ)
4	multiply	; АМАХ*TRUNC(ABS(A)/АМАХ)=Y
194	store M2	; Y - в ячейку M2
225	recall M1	; A, Y
49	duplicate	; A, A, Y
224	recall M0	; АМАХ, А, А, Y
10	>=	; A >= АМАХ ? Результат: 0 или 1
226	recall M2	; Y, [0 ИЛИ 1], А, Y.
4	multiply	; Y*(A>=АМАХ), А, Y.
3	subtract	; Y*(A>=АМАХ)-А, Y
56	end calc	; Выход из калькулятора.
6		; Теперь надо подготовить регистр В к
13	LD B,0B	; операции "<".
239	RST 28H	
224	recall M0	; АМАХ на стек
226	recall M2	; Y на стек
15	add	; АМАХ+Y
225	recall M1	; А на стек
160	const zero	; 0 на стек
13	<	; A < 0 ?
4	multiply	; (АМАХ+Y)*(A<0)
15	add	; A-Y*(A>=АМАХ)+(АМАХ+Y)*(A<0) - что и
		; требовалось...
1	exchange	; Лучше не оставлять на стеке "мусора".
2	delete	; В данной случае за результатом идет лиш-
		; нее число, оно удаляется.
56	end calc	; После чего - выход из процедуры.
		; Результат на стеке.
201	RET	

Апология Бейсика.

Вот и зашла речь о наболевшем вопросе: на каком языке программировать? Теперь, когда структура программы более-менее ясна, самое время порассуждать на эту тему.

Не стоит торопиться с ответом: "конечно же, в кодах". И вот почему. При всех бесспорных преимуществах - скорости, экономии памяти, возможности создавать нестандартные эффекты - даже отлаженная машинно-кодовая программа останется как бы замороженной. Изменить какие-либо детали еще можно, но структуру... легче писать заново, чем резать по живому. Если же автор серьезно интересуется астрологией, его взгляды на предмет не стоят на месте. То и дело возникает желание попробовать новую методику или изменить привычную. То и дело говоришь себе: "Вот этого я раньше не учитывал, а надо бы".

Хочу подчеркнуть, структура, которую я описал, подходит для скромной программы, не претендующей на то, чтобы удовлетворить неслабые потребности профессионального астролога. А методов, помимо гороскопа рождения - хоть пруд пруди...

Совершенствуешься и в умении решать вычислительные задачи. Я, например, совсем недавно узнал, как проще всего рассчитывать орбиту Плутона (это запутанный вопрос в астрономии, поскольку Плутон был открыт сравнительно недавно). И до сих пор не нашел

простого способа выполнять на восьмиразрядном "Спектруме" математические операции с двойной точностью, как, например, в GW-BASICe: задал "DEFDBL A-T" - и все переменные от А до Т становятся числами с удвоенной разрядностью. Но рано или поздно, ведь, узнаю, быть может не без помощи "ZX-РЕВЮ". Кстати, не занимался ли кто-нибудь из читателей этим вопросом? Не думаю, чтобы это было архисложной задачей. В компьютере MSX, построенном на том же процессоре Z-80, все числа по умолчанию 16-разрядные! Это очень пригодилось бы в календарных расчетах (9-я цифра в юлианской дате не обязательна, но ее присутствие повысило бы точность).

... Еще один "минус" кодовой программы: мониторы-отладчики, по крайней мере, те, что мне известны, не "берут" кодов калькулятора. Попробуйте сами и увидите такую фантазмагорию! Дабы уменьшить мучения, мне пришлось делать специальный отладчик; он позволял писать мнемониками и макро-командами, переводил числа в упакованную форму - вобщем, выполнил всякую рутину, но при всем этом был лишен удобств пакета GENS/MONS. Другое дело, игра - там калькулятор используется не так часто, задачи иные, а потому и подход не тот.

Недостатки стандартного БЕЙСИКа общеизвестны. Ладно, пускай, вычисления. Хуже то, что ради экономии памяти приходится идти на уловки, делающие порой программу нечитаемой. Отсутствуют некоторые команды, очень полезные для создания хорошей структуры, типа ON...GOTO, зато спектрумовский БЕЙСИК один из самых демократичных. Как-то я попробовал на пижонском компьютере MSX написать нечто вроде:

```
GOSUB(VAL"1000" AND X=SGN PI)+(VAL"2000" AND X=BIN).
```

Беднягу, кажется, стошнило...

Есть ли другие варианты? Первое, что приходит в голову - ПАСКАЛЬ, однако, повозившись недельку-другую с версией фирмы Hisoft, приходишь к выводу, что авторы ее заботились в первую очередь о том, как облегчить себе жизнь. Вещественные числа там имеют всего 6 значащих цифр (*5). Чтобы получить результат такой же точный, как на БЕЙСИКе, приходится проделывать разные арифметические трюки. Из-за этого и по другим причинам катастрофически не хватает памяти. Нет полной ясности относительно того, как Паскаль взаимодействует с системой - распространенные руководства говорят об этом крайне скупно. Так что далеко не все машинно-кодовые фрагменты, которые пытаешься вставить в программу, используя оператор INLINE - особенно, если в них есть обращения к ПЗУ, дают ожидаемые результаты.

Программу написать можно, в этом я убедился. Работать она будет исключительно быстро, читать ее текст и отлаживать - одно удовольствие - ПАСКАЛЬ есть ПАСКАЛЬ. Но считать она будет максимум в рамках двух столетий, делать сможет очень немного... и никаких красот. Может быть, есть смысл сделать на Паскале графический табель-календарь, показывающий, как планеты движутся в течение, скажем, заданного месяца - благо, разрешение экрана не настолько высоко, чтобы на графике были видны минуты, зато важно, чтобы расчеты выполнялись быстро.

С другой стороны, ничуть не худшего эффекта можно добиться с помощью TOBOS-FP, пожалуй, единственного из Бейсик-компиляторов, который оправдывает надежды. С ним пришлось немало повозиться. Часть того, что о нем написано - не только в "таблоидных" инструкциях, но и в такой серьезной книге, как "Диалекты Бейсика для ZX-SPECTRUM" - оказалось неточным. А кое о чем важном там вообще не сказано, например, о том, что объектный код является релоцируемым. Несмотря на то, что непосредственно после компиляции он располагается с адреса 24000, его можно загружать и запускать с другого адреса - скажем, с 26000, расположив ниже собственные кодовые подпрограммы. Лишь бы он не затирал рабочие процедуры и стек компилятора.

Напомню, что для компиляции больших программ в TOBOSe предусмотрен режим "компиляции с уничтожением исходного текста". Оказалось, этот режим не работает с дисковой системой, которая сдвигает БЕЙСИК вверх. Так что приходится инициализировать стандартную SINCLAIR-систему страшной командой RANDOMIZE USR 0, загружать с ленты компилятор и исходный текст, компилировать, записывать полученный объектный код на

ленту, потом загружать его с ленты в новый адрес и только после этого переписывать на диск (* /6). Удовольствия мало. Тем не менее, мне удалось откомпилировать таким образом довольно объемную программу, включавшую помимо расчетов еще и графику, сервис, разные виды таблиц - большую часть того, что необходимо профессиональному астрологу - и знакогенератор, печатающий 60 символов в строке. Больше всего поразила скорость. Хотите верьте, хотите нет, она была почти такой же, как у программы "PRIMA", написанной Б.Богдановым для IBM на Турбо-Паскале!

Однако, увы, за все приходится платить. Недостаток TOBOSa - пониженная точность расчетов (он работает с 7-значными числами); памяти, несмотря на все, он съедает много: хочешь добавить в программу какой-нибудь милый пустячок - не тут-то было...

Вариант, который мне кажется почти идеальным для некоммерческой астрологической программы - это БЕТА-БЕЙСИК. О его преимуществах было достаточно написано в "ZX-РЕВЮ". Он как будто нарочно создан для подобных программ. Позже пойдет речь об уравнении Кеплера. Оно очень красиво вписывается в структуру DO...UNTIL. Оператор ON...GOSUB в паре мест просто необходим. Функции SINE, COSE, знакогенератор 4/7 пикселей, возможность гибко манипулировать сеткой графических координат, окна - все это помогает наглядно иллюстрировать результаты.

Параметры орбит.

Я уже писал, что для расчета координат небесных тел нужно знать во-первых, период T , а во-вторых, параметры их орбит в начальную эпоху T_0 . Для каждой из планет таких параметров 6: средняя аномалия, эксцентриситет, большая полуось, долгота перигелия, долгота восходящего узла и наклон орбиты. Обозначим их переменными ZM , ZE , ZA , ZW , ZQ , ZI .

Допустим, дата у нас 24 июня 1976 г. Время - 0 ч. по Гринвичу. Тогда юлианская дата:

$JD = 2443683.5,$

a

$T = (2443683.5 - 2415020) / 36525 = 0.78476386.$

Найдем среднюю аномалию ZM для Меркурия. В таблице даны три числа:

102,27938,

149472,52,

$7E-6.$

Обозначим их $X1$, $X2$, $X3$.

$ZM = X1 + X2 * T + X3 * T * T$

(или $FN P(X1, X2, X3, T) = 117402.91$ градусов = 42.91004 градусов = $0,74892085$ радиан.

Точно так же находим остальные значения.

Все величины, кроме ZA (астрономические единицы) и ZE нужно перевести в радианы!

Поскольку каждая орбита представляет собой эллипс, данные параметры описывают именно эту геометрическую фигуру, а также ее положение относительно орбиты Земли.

Положения Солнца и планет.

Средняя аномалия ZM показывает, где бы находилась планета, если бы она двигалась равномерно по кругу. Зная ее и эксцентриситет эллипса, можно найти эксцентрическую аномалию EA и истинную аномалию ZV , т.е. перевести планету на эллипс. Для определения EA служит уравнение Кеплера:

$EA = ZE * SIN(EA) = ZM.$

Очевидно, решить его не так-то просто, Но компьютеру все нипочем! Вот как это делается.

197 REM HELI01

198 REM

199 REM Уравнение Кеплера. На входе ZE,ZM. На выходе EA

200 LET EA=ZM: REM начальное значение EA

202 LET EPSYLON=1E-6: REM задаем точность

205 REM начало цикла

```

210 LET DELTA=EA-ZE*SIN(EA)-ZM: REM невязка.
220 LET EA=EA-DELTA/(1-ZE*COS(EA)): REM очередное значение EA
230 IF ABS(DELTA)>EPSYLON THEN GOTO 210: REM к началу цикла, если не достигнута требуемая
    точность

```

А вот как то же самое выглядит в кодах калькулятора (к моменту вызова этой процедуры на вершине калькуляторного стека должно находиться ZM, за ним - ZE. Сама процедура может быть расположена в любом месте оперативной памяти). См. листинг-2.

Листинг-2

Дес. код	Мнемоника	Комментарий
239	RST28H	; Вкл. калькулятора
195	store M3	; Начальное значение EA, т.е. ZM - ; в 3-ю ячейку памяти
195	store M4	; ZM в 4-ю ячейку.
1	exchange	; Меняем местами ZM и ZE
197	store M5	; ZE в 5-ю ячейку

2	delete	; Удаляем верхнее число со стека. ; Это начало цикла.
49	duplicate	; Копирование вершины стека.
229	recall M5	; Вызываем ZE
1	exchange	; Теперь стек выглядит так: EA, ZE, EA
31	sin	; sin(EA), ZE, EA
4	multiply	; ZE*sin(EA), EA
3	subtract	; EA-ZE*sin(EA)
226	recall M4	; вызываем ZM
3	Subtract	; EA-ZE*sin(EA)-ZM, т.е. невязка DELTA.
49	duplicate	; DELTA, DELTA
161	const one	; 1, DELTA, DELTA
227	recall M3	; EA на стек
32	cos	; COS(EA)
229	recall M5	; ZE на стек
4	multiply	; cos(EA)*ZE
3	subtract	; 1-cos(EA)*ZE, DELTA, DELTA
5	divide	; DELTA/(1-cos(EA)*ZE), DELTA
227	recall M3	; EA на стек
1	exchange	;
3	subtract	; EA минус последний результат. Получено ; очередное значение EA
195	store M3	; сохраняем его в 3-й ячейке
1	exchange	; меняем местами EA и DELTA
42	abs	; abs(DELTA)
52		; Это ввод
221		; числа
6		; 1E-5 в упакованной форме
55		; (допустимая
189		; величина
5		; невязки)
3	subtract	; abs(DELTA)-1E-6
55	gt zero	; >0?
0		
225	jump true-31	Если да, то переход к началу цикла

56	end calc.	; Выключение калькулятора.
201	RET	; Результат на вершине стека.

Теперь можно найти истинную аномалию ZV:

```

235 REM истинная аномалия. На входе EA, ZE, ZA (большая полуось).
240 LET A=ZA*SIN(EA)*SQR(1-ZE*ZE)
250 LET B=ZA*COS(EA)-ZE)
260 GOSUB 140: REM к процедуре ARCTANGENS
270 LET ZV=X

```

Затем радиус-вектор, ZR:

```
275 REM радиус-вектор. На входе ZA, EA, ZE
290 LET ZR=ZA*(1-ZE*COS(EA))
```

Таким образом, величины ZR, ZV найдены, что и было намечено в первоначальной схеме. На этом можно было бы и закончить подпрограмму HELIO1, однако, здесь же удобнее вычислить еще одну промежуточную величину, используя очередной параметр орбиты. Обозначим ее ZU.

```
290 LET A=ZV+ZW: LET AMAX=PI+PI
300 GOSUB 100: REM к процедуре INTERVAL
310 LET ZU=X
320 RETURN: REM конец подпрограммы HELIO1
```

Долготу солнца (SL) найти теперь просто. В случае с ним ZU -это долгота Земли. Так что $SL=ZU+PI$. Полученную величину надо "прогнать" через процедуру INTERVAL. С планетами сложнее. Вот как выглядит расчет для них.

```
321 REM HELIO2
322 REM гелиоцентрические координаты планет.
323 REM на входе ZQ,ZI,ZU, на выходе ZL, ZB - гелиоц. широта и долгота
324 REM
325 REM *** ZI.
330 LET A=COS(ZI)*sin(ZU):LET B=COS(ZU)
340 GOSUB 140: REM к процедуре ARCTANGENS
350 LET A=X+ZQ: LET AMAX=PI+PI
360 GOSUB 100: REM к процедуре INTERVAL
370 LET ZL=X
380 LET ZB=ASN(SIN(ZU)*SIN(ZI))
390 RETURN
395 REM GEOCENTRIC
396 REM переход от гелио- к геоцентрическим долготе и широте (LO,LA)
397 REM на входе: ZR, ZL, ZB, а также SR, SL - радиус-вектор и долгота Солнца.
398 REM Z, A, M, C, N, D - локальные переменные
399 REM
400 LET A=ZL-SL:LET M=ZR*COS(ZB):LET C=ZR*SIN(ZB)
410 LET N=M*SIN(A): LET D=M*COS(A)+SR
415 REM *** LO
420 LET A=N: LET B=D: GOSUB 140:REM ARCTANGENS
425 LET A=X+SL: LET B=PI+PI:GOSUB 100: REM INTERVAL
430 LET LO=X
435 REM *** расстояние от Земли
440 LET Z=SQR(M*N+D*D+C*C)
445 REM *** LA
450 LET LA=ASN(ZR/Z*SIN(ZB))
460 RETURN
```

Осталось перевести LO и LA из радиан в градусы и передать массиву A().

Луна.

Вместе с Солнцем и Асцендентом, Луна - важнейший показатель в любом гороскопе, а в женском часто самый главный. Достаточно вспомнить, с чем она связывалась в представлениях разных народов, чтобы понять ее значение в астрологии. В древних религиях ее почитаем, как богиню зачатий и материнства. Если Солнце - активное, мужское начало, то Луна - уравновешивающая его пассивная, женская сторона природы, инстинкт самосохранения, чувства... Известно, что лунные циклы оказывают влияние на многие процессы в животном и растительном мире, на здоровье человека. Древние славяне лечились, выполняли земледельческие работы и даже солили капусту сообразно с определенными фазами Луны.

Луна движется через Зодиак быстро, проходя полный круг за месяц, а один знак -

примерно за двое суток. Точно вычислить ее положение в заданный момент времени непросто - слишком много факторов приходится учитывать. На каждом шагу надо вносить поправки. Целиком все занимает пару книжных страниц. Тем, кто желает вычислить положение Луны с точностью до минуты, рекомендую обратиться к книге Ж.Месса "Астрономические формулы для калькуляторов". Здесь же приведу упрощенный расчет, занимающий немного меньше места и дающий точность плюс-минус 10 минут дуги. Это не так страшно, как может показаться на первый взгляд - ведь видимый диск Луны "растянут" на целых 30 минут, полученная же долгота указывает положение центра диска.

```

495 REM MOON
496 REM на входе T,JD,SL,SN. На выходе X,Y,ZQ - долгота, широта Луны и долгота Лунного узла
498 REM C,D - вспомогательные локальные переменные
499 REM параметры орбиты
500 LET D=JD-2415020: REM число дней, прошедших от начальной эпохи
505 DIM P(5): REM для поправок
510 LET ZL=270.434164+D*13.176397:REM средняя долгота
520 LET ZW=334.32956+D*0.11140408: REM долгота перигея
530 LET ZM=ZL-ZW: REM средняя аномалия
540 LET ZQ=259.18328-D*0.052953923 REM долгота узла
543 LET ZI=5.145396: REM наклон орбиты
545 REM приведение параметров в интервал 0...360 и перевод в радианы.
550 LET AMAX=350
500 LET A=ZM: GOSUB 100:LET ZM=X*(PI/180)
570 LET A=ZL: GOSUB 100:LET ZL=X*(PI/180)
580 LET A=ZQ: GOSUB 100:LET ZQ=X*(PI/180)
590 LET ZI=ZI*(PI/180)
595 REM расчет основных поправок
600 LET C=ZL-SL
610 LET P(1)=1.2739*SIN(C+C-M)
620 LET P(2)=0.1858*SIN(SM)
630 LET P(3)=0.37*SIN(SM)
640 LET ZM=ZM+(P(1)-P(2)-P(3))*(PI/180)
650 LET P(4)=6.2686*SIN(ZM)
660 LET P(5)=0.214*SIN(ZM+ZM)
670 LET ZL=ZL+P(1)+P(4)-P(8)+P(5))*(PI/180)
680 LET ZL=ZL+0.6583*SIN(2*ZL-SL))*(PI/180)
690 LET ZQ=ZQ-0.16*SIN(SM)*(PI/180)
700 LET C=ZL-ZQ
705 REM *** долгота
710 LET A=SIN(C)*COS(ZI): LET B=COS(C)
720 GOSUB 140: REM ARCTANGENS
725 LET A=X+ZQ: LET AMAX=PI+PI: GOSUB 100: REM INTERVAL
730 REM *** широта
740 LET Y=ASN(SIN(C)*SIN(C)*SIN(ZI))
750 LET X=X/(PI/180): LET Y=Y/(PI/180): LET ZQ=ZQ/(PI/180)
760 RETURN

```

Разумеется, вместо (PI/180) лучше использовать константу S, которая равна 1.7453292E-2, т.е. приблизительно 1 радиану. А массив P(5) объявить в начале программы.

Поправки на возмущения планетных орбит.

Больше всего отклоняются от правильного в геометрическом смысле движения Сатурн и Юпитер. К их средним долготам (ZL) надо прибавить:

Для Юпитера:

```

((0.3314-0.0103*P(1))*SIN(P(4))-
0.0644*P(1)*COS(P(4)))*(PI/180)

```

Для Сатурна:

```

((0.1609*P(1)-0.0105)*COS(P(4))+
(0.0182*P(1)+0.8142)*SIN(P(4))-
0.1488*SIN(P(5))-0.0408*SIN(P(5)+
P(5))+0.0856*SIN(P(5))*COS(P(3))+
0.0813*COS(P(5))*SIN(P(3)))*(PI/180).

```

где:

$$P(1)=T/5+0.1$$

$$P(2)=(237.47555+3034.9061*T)*(PI/180)$$

$$P(3)=(265.91650+1222.1139*T)*(PI/180)$$

$$P(4)=5*P(3)-2*P(2) \quad P(5)=P(3)-P(2)$$

Как всегда, полученные величины лучше привести в интервал $0...PI*2$.

Расчет астрологических домов по системе В.Коха.

```
885 REM
886 REM KOCH HOUSES
887 REM
888 REM на входе: GMT-время по Гринвичу в часах и десятых долях часа, L0 - долгота в
      градусах, "-" к востоку. LA-широта в градусах, "-" к югу, EC- наклон эклиптики в
      градусах, T. Выход: массив H(1...12) - эклиптическая доля гота границ домов.
892 REM
893 REM Прямое восхождение меридиана
894 REM
900 LET RAMC=FN P(6.6460656,2400.0513,2.581E-5,T):REM POLYNOME
910 LET A=(RAMC+GMT)*15-L0
920 LET LA1=LA*(PI/180)
930 IF LA>(90-EC) THEN LET A=RAMC+180: LET LA1=-LA1
940 LET A=RAMC: LET AMAX=360:GOSUB 100: LET RAMC=X: REM INTERVAL
945 LET RAMC=RAMC*(PI/180): LET EC1=EC*(PI/180):REM перевод в радианы
950 REM середина неба
960 LET A=SIN(RAMC):LET B=COS(RAMC)*COS(EC1): GOSUB 140: REM ARCTANGENS
965 LET MC=X
970 REM Асцендент
980 LET A=COS(RAMC):LET B=-(SIN(EC)*TAN(LA1)*COS(EC)*SIN(RAMC):GOSUB 140
985 LET AS=X
990 LET L=1: LET X=AS:GOSUB 1140
993 LET L=10: LET X=MC:GOSUB 1140
995 REM Вершины домов
1000 LET X0=SIN(EC1)*TAN(LA1)/COS(EC1)*SIN(RAMC)
1010 RESTORE 1020
1020 DATA 0.52350879,1.0471976,2.0943952,2.617994
1030 LET X4=-3
1040 FOR J=1 TO 4
1050 READ X: LET X4=X4+1
1060 IF X4=0 THEN LET X4=1
1070 GOSUB 1100
1080 NEXT J
1090 RETURN
1095 REM вычисления
1100 LET X3=RAMC+X+ATN(X0/SQR(1-X0*X0))*X4/3
1110 LET A=SIN(X3):LET B=COS(X3)*COS(EC1)-TAN(LA1)*SIN(EC1)
1120 LET L=L+1: IF L>12 THEN LET L=L-11
1130 GOSUB 140: REM ARCTANGENS
1140 LET H(L)=X/(PI/180):LET A=X+PI: LET AMAX=PI+PI:GOSUB 100: REM INTERVAL
1150 LET H(L+6-12*(L>7))=X/(PI/180): REM противоположный дом
1160 RETURN
```

Управляющая программа.

Поскольку для ряда простых расчетов я давал лишь формулы, то нет смысла подробно описывать главный блок, показывая номера строк для переходов, важнее сам алгоритм. Выглядит он так:

1) Ввести исходные данные. Проверить, не допустил ли пользователь ошибку. Если да, то начать сначала. Распечатать эти данные на экране и спросить у него самого, все ли в порядке. Если нет, начать сначала. Перевести все в формат, который будет использовать программа.

2) Рассчитать юлианскую дату, соответствующую заданной дате и времени, вычислить период T.

3) Рассчитать по T и данным из таблицы исходные параметры орбиты Солнца: ZM, ZE,

ZA, ZW. Найти положение Солнца (см. HELIO1). Сохранить среднюю долготу, радиус-вектор и среднюю аномалию Солнца в особых переменных.

4) Найти положение Луны и Северного Лунного узла - $M(1,1)$, $M(2,1)$, $M(11,1)$, $M(12,1)$

5) Вычислить значения $P(1..5)$, которые понадобятся при расчете поправок на долготы Сатурна и Юпитера

6) Для планет:

Цикл для J от 2 до 9

Определить ZM, ZE, ZA, ZW, ZQ, ZI.

Найти гелиоцентрические координаты. Если J=5 или 6 (Юпитер, Плутон), вычислить поправки и прибавить их к ZL.

Перейти к геоцентрическим координатам.

Перевести результат в градусы и присвоить массиву M().

7) Найти наклон эклиптики EC.

8) Рассчитать границы домов H().

9) Вывести результат на дисплей или принтер.

10) Вернуться к началу.

Я намеренно не касался пункта (9) только потому, что тема эта слишком обширна. Сюда входят вопросы дизайна и графики, форматирования чисел... возможно даже несколько нестандартного прерывания, если Вы желаете, чтобы в углу экрана имелись часы - для всего этого потребовалась бы отдельная статья. Хотелось бы только обратить внимание читателей на следующее: все результаты, хранящиеся в массивах H() и M() - вещественные числа, долготы лежат в диапазоне 0...360 градусов. Выводить на экран лучше градусы и минуты. Если же придерживаться астрологической традиции, то надо указывать знак Зодиака. Так, например, 338.25002 преобразуется в строку "08:04 Рыб". Один из возможных способов вывода представлен на рис.3 (получен из зарубежной программы фирмы Protheus software).

В заключение прошу извинить меня за слишком поверхностное изложение астрономии. Всем, кому такой уровень покажется недостаточным, горячо рекомендую книги: Ж.Меес - "Астрономические формулы для калькуляторов", М., Мир, 1988; П. Дэффет-Смит - "Практическая астрономия с калькулятором", М., Мир, 1982. Что же касается тонкостей астрологии, то среди моря доступной сегодня литературы я бы выделил учебники Сефариала, Джорджа Левеллина, Ф. Сакоян, Хет Монстера и философские работы Редиара.

* * *

КОММЕНТАРИИ "ИНФОРКОМа".

/1. "Эклиптика" - правильнее говорить - "плоскость эклиптики".

Это плоскость, которая совпадает с плоскостью орбиты земли. Поскольку Земля вращается вокруг Солнца, то естественно, что Солнце всегда лежит в этой плоскости.

/2. Геоцентрические координаты - это координаты в системе, в которой начало отсчета привязано к центру Земли. Эклиптические геоцентрические координаты - координаты в системе, в которой начало координат совпадает с центром Земли, а главная плоскость XY совпадает с плоскостью эклиптики.

/3. Поскольку Солнце естественно всегда лежит в плоскости эклиптики, то его широта в эклиптической геоцентрической системе координат всегда равна нулю.

/4. Гелиоцентрическая система координат - в отличие от геоцентрической связана не с Землей, а с Солнцем. Центр этой системы находится в центре Солнца, а главная плоскость XY - совпадает с плоскостью эклиптики.

/5. Это действительно так, но надо признать, что это не столько облегчает жизнь создателям ПАСКАЛЯ, сколько пользователям. Ведь благодаря этому в ПАСКАЛе HP4T удается использовать только четыре восьмиразрядных регистра для хранения действительных чисел (вместо пяти). Это значительно упрощает интерфейс программ на

ПАСКАЛе с собственными процедурами пользователя в машинном коде. Из м/к легко находятся и эксплуатируются переменные ПАСКАЛЯ.

/6. По-видимому, используя информацию, приведенную в статье Алексева А.Г. "Секреты TRDOS" (см. ZX-РЕВЮ N1,2 за 1993 г.) этот недостаток компилятора можно было бы и устранить. Не исключено, что кто-то это уже и сделал.

Справочные таблицы для расчетов элементов орбит планет.

SUN

ZM	358.47583	35999.05	-.00015
ZE	.01675104	-.0000418	-1.26E-7
ZA	1.0000002	0	0
ZV	101.22083	1.71918	.00045

MERCURY

ZM	102.27938	149472.52	7E-6
ZE	0.20561421	.00002046	-3E-8
ZA	0.3870986	0	0
ZV	28.753753	0.3702806	.0001208
ZQ	47.145944	1.1852083	.0001739
ZI	7.002881	.0018608	-.0000183

VENUS

ZH	212.60322	58517.804	.001286
ZE	.00682069	-.00004774	9.1E-8
ZA	0.7233316	0	0
ZU	54.384186	0.5081861	-.0013664
ZQ	75.779647	0.89985	.00041
ZI	3.393631	.0010058	-1E-6

MARS

ZM	319.51913	19139.855	.000181
ZE	.0933129	.000092064	-7.75-8
ZA	1.5236883	0	0
ZW	285.43176	1.0697667	.0001313
ZQ	48.786442	0.7709917	-1.4E-6
ZI	1.850333	-.000675	.0000126

JUPITER

ZM	225.32833	3034.692	-.000722
Z8	.04833475	.00016418	-4.876E-7
ZA	5.2025613	0	0
ZV	273.27756	0.5994317	.00070405
ZQ	99.443414	1.01053	.00035222
ZI	1.308736	-.0056961	3.9E-6

SATURN

ZM	175.46622	1221.5515	-.000502
ZE	.05589232	-.0003455	-7.28E-7
ZA	9.554747	0	0
ZV	338.3078	1.0852207	.00097854
ZQ	112.79041	0.8731951	-.00015218
ZI	2.492519	-.0039189	-.00001549

URANUS

ZM	72.64878	428.37911	.000079
ZE	.0463444	-.00002658	7.7E-8
ZA	19.21814	0	0
ZW	99.071581	0.985765	-.0010745
ZQ	73.477111	0.4986678	.0013117
ZI	0.772464	.0006253	.0000395

NEPTUNE

ZM	37.73063	218.46134	-.00007
ZE	.00899704	6.338-6	-2E-9
ZA	30.10957	0	0
ZW	276.04597	0.3256394	.00014095
ZQ	130.68139	1.098935	.00024987
ZI	1.779242	-.0095436	-9.1E-6

PLUTO

ZM	229.94722	144.91306	0
ZE	.24864	0	0
ZA	39.51774	0	0
ZW	113.52139	0	0
ZQ	108.95444	1.39601	.00031
ZI	17.14678	0	0

СДЕЛАЙТЕ САМИ

(C) Т.Д. Фрост (T.D. FROST), 1986.

(C) Адаптация, Алексеев А.Г., 1993.

(C) Авторизованный перевод, редактирование, "ИНФОРКОМ". 1993 г.

ADVENTURE GAMES

Введение.

Те, кто интересуются адвентюрными играми, знают о существовании программ для автоматизированного создания таких игр - QUILL, GAC и др. Интерес к этим программам поддерживается прежде всего тем, что очень часто любителям игровых программ приходят в голову отличные сюжеты, рождаются готовые сценарии и человек хочет сам создать карту своего выдуманного мира, населить пещеры троллями, попрыгать золото в сундуках и насытить игру хитроумными головоломками. Что делать в такой ситуации? Если Вы не программист, то дело ваше почти безнадежное.

Но на помощь Вам может прийти "ABS" - это система, которая позволит сделать первый шаг. Она может пригодиться и опытным программистам, ведь для того, чтобы проверить свою задуманную концепцию вовсе не нужно садиться и писать игровую программу. На это могут уйти месяцы, а окажется, что концепция не интересна, сценарий скучен и все труды затрачены зря. Гораздо удобнее иметь такой гейм-дизайнер, как ABS и тогда проверить свою концепцию можно в течение нескольких часов.

Эта программа будет полезна и тем, кто просто интересуется вопросом "Как это делается?" Как устроены адвентюрные игры, как они работают - все это Вы узнаете, разобрав эту систему.

"ABS", если вдуматься, готовый инструмент для создания обучающих программ для детей типа "Путешествие Васи Кляксина в страну Невыученных Уроков". Когда наберете необходимый навык, сможете делать такие программы своим детям хоть по пять штук в неделю. Вы можете также быстро создать игру-иллюстрацию к той книжке, которую только что прочитал Ваш ребенок или ученик.

И, наконец, нет предела творческому воображению наших читателей - опытные хакеры смогут развить и доработать эту программу - сократить ее размер, сделать генерируемые программы более насыщенными и оживленными, более "приспособленными" для русского языка - дерзайте, творите.

Если дело пойдет, то мы с удовольствием поддержим его развитие и в разделе "Возвращаясь к напечатанному" будем сообщать о новых идеях и приемах, открытых нашими читателями.

Структура системы.

Система ABS довольно объемна. Листинг 1 - это сама программа ABS. Листинги 2...5 - демонстрационная адвентюрная программа, показывающая, как пользоваться программой ABS. В них содержится вся необходимая для корневой программы информация - количество комнат, количество объектов, описания комнат и т.п. Под "комнатами" здесь понимаются конечно не физические комнаты, а локации, т.е. участки, из которых состоит игровое пространство - это может быть и лес и горы и борт космического корабля и действительно комнаты.

Во время работы демонстрационная программа выдает на экран описание текущей локации (а может выводиться графика, если она есть) и список объектов, видимых в данном месте. Программа задает вопрос: "ВАШИ ДЕЙСТВИЯ?" и ждет от Вас команды. Ваша команда типа: "ВЗЯТЬ ФАКЕЛ" печатается на экране, затем анализируется сначала глагол, потом существительное и исполняется программой.

Головная программа (Листинг 1) содержит кодировку часто встречающихся действий,

например "С" или "ИДТИ С" означает "ИДТИ НА СЕВЕР".

Листинги.

Порядок работы с листингами, их ввода и их выгрузки на ленту - довольно сложен, но если Вы аккуратно будете следовать всем указаниям, то сможете не только установить головной блок, но и ввести и запустить демонстрацию.

Программа русифицирована нашим традиционным способом с использованием утолщенного русско-латинского символьного набора в кодировке ASCII КОИ-7, сформированного в файл "chr" CODE.

Сначала введите Листинг_1. Он сохраняется на ленте прямой командой RUN 9990. Это "Мастер-копия" головной программы. Вы будете ею пользоваться всякий раз, когда захотите создать новую игру, а пока спрячьте эту кассету в надежном месте, сейчас она нам больше не понадобится.

Листинг_1. Программа "ABS". Мастер-копия. Файл "ABS".

```
1 GO TO 100
2 CLEAR 49999: LOAD "chr" CODE 50000
4 GO TO 0
5 GO TO 9990
8 POKE 23606,80: POKE 23607,194: RETURN
9 POKE 23606,0: POKE 23607,60: RETURN
40 CLS : POKE 33658,8: GO SUB 8
42 PRINT PAPER 0; INK 6; BRIGHT 1; AT 1,3;" ADVENTURE BUILDER SYSTEM "
44 RETURN
50 PRINT INK 2; BRIGHT 1; FLASH 1; AT 8,6;"ПОДОЖДИТЕ НЕМНОГО..."; AT 10,5;"ИДЕТ ГЕНЕРАЦИЯ
  ДАННЫХ."
52 RESTORE 50: FOR X=64008 TO 64047: READ A: POKE X,A: NEXT X
54 DATA 254,226,40,15,254,195,40,15,254,204,40,15,254,203,40,15,195,48,250,62,126,24,10,62,
  124,24,6,62,123,24,2,62,125,195,72,252,121,222,133,201
58 RESTORE 60: FOR X=64430 TO 65077: READ A: POKE X,A: NEXT X
60 DATA 205,190,251,62,21,50,60,250,24,33,58,137,92,254,2,192,62,3,50,137,92,229,6,0,205,0,
  14,225,201,58,137,92,71,62,24,144,254,22,40,216
61 DATA 50,60,250,33,61,250,6,55,54,0,35,16,251,33,61,250,54,1,35,54,1,35,54,0,33,106,92,54,
  0,62,22,215,58,60,250,215,62,0,215,62
62 DATA 62,215,62,22,215,58,60,250,215,58,61,250,215,60,50,61,250,62,42,215,1,0,0,205,61,31,
  205,191,2,33,8,92,78,62,13,185,202,181,252,62
63 DATA 12,185,202,132,252,62,8,185,202,132,252,58,61,250,254,32,202,48,250,121,254,32,40,
  13,254,230,212,44,250,254,128,210,8,250,79,24,9,79,58,61,250
64 DATA 254,3,218,48,250,65,62,22,215,58,60,250,215,58,61,250,61,215,60,50,61,250,120,245,
  215,241,42,62,250,17,63,250,25,119,42,62,250,35,34
65 DATA 62,250,17,50,0,33,200,1,205,181,3,195,243,251,58,61,250,254,3,218,48,250,61,50,61,
  250,62,22,215,58,60,250,215,58,61,250,215,61,50,61,250
66 DATA 62,32,215,42,62,250,1,62,250,9,54,0,58,62,250,61,50,62,250,195,120,252,58,61,250,
  254,3,218,48,250,62,22,215,56,60,250,215,58,61,250
67 DATA 61,215,62,32,215,17,25,0,33,250,0,205,181,3,33,63,250,54,32,35,229,6,31,126,254,48,
  56,3,35,16,248,225,17,100,250,52,31,144
68 DATA 6,0,184,250,251,252,126,18,35,19,16,250,24,17,79,71,126,18,35,19,16,250,62,0,145,71,
  175,19,18,16,252,33,93,250,6,31,126,254,0,32
69 DATA 3,43,16,248,126,254,32,40,3,43,16,248,5,72,6,0,33,64,250,9,6,0,17,107,250,126,18,35,
  19,16,250,33,0,0,17,100,250,6,0,26,190
70 DATA 32,10,35,19,16,248,126,50,114,250,24,17,72,6,0,9,35,62,199,190,40,2,24,225,33,114,
  250,54,200,33,0,0,17,107,250,6,0,26,190,32
71 DATA 10,35,19,16,248,126,50,115,250,24,17,72,6,0,9,35,62,199,190,40,2,24,225,33,115,250,
  54,201,58,115,250,71,62,0,184,216,33,24,251,58
72 DATA 115,350,6,0,79,9,62,1,190,192,58,115,250,6,0,128,50,115,250,201,33,0,0,6,0,54,0,35,
  16,251,17,0,0,6,0,58,24,251,33,124,251
73 DATA 190,40,4,35,16,250,201,79,62,0,144,18,19,35,121,24,243,33,0,0,6,0,54,0,35,16,251,17,
  0,0,6,0,62,99,33,124,251,190,40
74 DATA 4,35,16,250,201,79,62,0,144,18,19,35,121,24,243,17,7,0,33,0,0,6,0,58,24,251,190,40,
  3,25,16,250,58,115,250,71,35,16,253,126
```

```

75 DATA 50,115,250,201,33,24,251,17,130,250,1,150,0,237,176,201,33,130,250,17,24,251,1,150,
    0,237,176,201,33,24,251,6,150,54,0,35,16,251,1,24,251
76 DATA 62,0,2,33,99,0,9
80 RESTORE 80: FOR X=64048 TO 64059
82 READ Z: POKE X,Z: NEXT X
84 DATA 17,5,1,33,125,1,205,181,3,195,18,252
89 RETURN
100 BORDER 7: PAPER 7: INK 0: CLS : POKE 23658,8: GO SUB 8
105 GO SUB 40: GO SUB 50: GO SUB 40
106 FOR X=1 TO 11: BEEP .1,X: NEXT X
107 FOR X=10 TO 1 STEP -1: BEEP .1,X: NEXT X
110 PRINT AT 2,0;"
    ВВЕДИТЕ ПОЛНОЕ КОЛИЧЕСТВО ГЛА-
    ГОЛОВ, ИСКЛЮЧАЯ:
    ИДТИ, ВЗЯТЬ (БРАТЬ), ПОЛОЖИТЬ
    (БРОСИТЬ), ПРОВЕРЯТЬ, СМОТРЕТЬ,
    ИНВЕНТАРЬ (И), СТОП (КОНЕЦ), СО-
    ХРАНИТЬ, ЗАГРУЗИТЬ;
    А ТАКЖЕ БУКВ, ЗАДАЮЩИХ НАПРАВЛЕ-
    НИЕ: С(ЕВЕР), Ю(Г). В(ОСТОК),
    З(АПАД), (ВВЕР)Х, (В)Н(ИЗ) - ВСЕ
    ЭТО УЖЕ ВВЕДЕНО С ПОМОЩЬЮ КОДОВ:"
115 PRINT TAB 8 "ИДТИ.....0"
116 PRINT TAB 8 "ВЗЯТЬ.....1"
117 PRINT TAB 8 "ПОЛОЖИТЬ...2"
118 PRINT TAB 8 "ПРОВЕРИТЬ..3"
119 PRINT TAB 8 "СМОТРЕТЬ...4"
120 PRINT TAB 8 "ИНВЕНТАРЬ..5"
121 PRINT TAB 8 "КОНЕЦ.....6"
122 PRINT TAB 8 "СОХРАНИТЬ..7"
123 PRINT TAB 8 "ЗАГРУЗИТЬ..8"
125 INPUT V
126 LET V=V+28
127 GO SUB 40
128 PRINT INK 7; PAPER 2; BRIGHT 1; AT 5,2; "СИМВОЛЫ ИДЕНТИФИКАЦИИ КОМАНДЫ"
129 PRINT AT 8,0;"
    ВВЕДИТЕ КОЛИЧЕСТВО СИМВОЛОВ,
    КОТОРЫЕ СЛУЖАТ ДЛЯ СРАВНЕНИЯ
    ВВЕДЕННОГО ГЛАГОЛА С ОРИГИНАЛОМ."
130 PRINT INK 2; AT 13,4; "ВВЕДИТЕ ЧИСЛО ОТ 3 ДО 7"
131 INPUT VL
132 IF VL<3 OR VL>7 THEN BEEP .5,20: GO TO 127
133 LET VP=VL+1: LET VPP=VP+1
134 LET DAV=64000-(V*VP)
140 GO SUB 40
150 PRINT AT 8,0;"
    ВВЕДИТЕ КОЛИЧЕСТВО ОБ'ЕКТОВ,
    КОТОРЫЕ БУДУТ ВХОДИТЬ В ИНВЕН-
    ТАРЬ ИГРЫ:"
155 INPUT MOB
156 IF MOB>50 THEN PRINT INK 2; AT 12,1; "ИЗВИНИТЕ, НО ЭТО СЛИШКОМ МНОГО!": BEEP .5,20: PAUSE
    50: GO TO 140
160 GO SUB 40
161 PRINT INK 7; PAPER 1; BRIGHT 1; AT 5,2; "СИМВОЛЫ ИДЕНТИФИКАЦИИ ОБ'ЕКТА"
162 PRINT AT 8,0;"
    ВВЕДИТЕ КОЛИЧЕСТВО СИМВОЛОВ,
    ПО КОТОРЫМ СУЩЕСТВИТЕЛЬНОЕ СРАВ-
    НИВАЕТСЯ С ОРИГИНАЛОМ"
163 PRINT INK 1; AT 13,4; "ВВЕДИТЕ ЧИСЛО ОТ 3 ДО 7"
164 INPUT NL
165 IF NL<3 OR NL>7 THEN BEEP .5,20: GO TO 160
166 LET NP=NL+1: LET NPP=NP+1
170 GO SUB 40
171 PRINT AT 8,0;" ВВЕДИТЕ МАКСИМАЛЬНУЮ ДЛИНУ ОПИСАНИЯ ОБ'ЕКТА:"
175 INPUT L

```

```

176 IF L>32 THEN PRINT INK 2;AT 12,1;"МАКСИМАЛЬНАЯ ДЛИНА - ТОЛЬКО 32!": ВЕР .5,20: PAUSE
    50:GO TO 170
179 LET AYAS=0
180 GO SUB 40
190 PRINT AT 8,0;"
    ВВЕДИТЕ ЧИСЛО ОБ'ЕКТОВ, КОТО-
    РЫЕ МОГУТ ИЗМЕНЯТЬ СВОЕ ОПИСАНИЕ
    ПО ХОДУ ИГРЫ, НАПРИМЕР:" INK 1'TAB 5;"ФАКЕЛ - ГОРЯЩИЙ ФАКЕЛ" INK 0'"ЕСЛИ ЭТО НЕ
    ТРЕБУЕТСЯ, ВВЕДИТЕ 0"
195 INPUT MCOB
196 IF AYAS=0 AND MCOB>7 THEN PRINT INK 2; AT 19,5;"ВЫ АБСОЛЮТНО УВЕРЕНЫ?": ВЕР .5,20:
    PAUSE 50: LET AYAS=1: GO TO 180
200 GO SUB 40
210 PRINT AT 8,0;"
    ВВЕДИТЕ ЧИСЛО ПРОЧИХ СЛОВ,
    КОТОРЫЕ ПО ВАШЕМУ ЖЕЛАНИЮ ДОЛЖНЫ
    РАСПОЗНАВАТЬСЯ ПРОГРАММОЙ:"
215 INPUT MON
220 LET N=MOB+MON+22
225 LET DAN=DAV-(N*NP)-5
230 GO SUB 40
235 PRINT AT 8,0;"
    ВВЕДИТЕ МАКСИМАЛЬНОЕ ЧИСЛО
    ОБ'ЕКТОВ, КОТОРЫЕ ОДНОВРЕМЕННО
    МОЖНО ВЗЯТЬ:"
240 INPUT MAX
245 GO SUB 40
255 PRINT AT 8,0;"ВВЕДИТЕ КОЛИЧЕСТВО ЛОКАЦИЙ:"
260 INPUT NOR
265 GO SUB 40
270 PRINT PAPER 2; INK 7; BRIGHT 1;AT 5,8; "ФАКТОР СКРОЛЛИНГА"
272 PRINT AT 8,0;"
    ВЫ МОЖЕТЕ СДЕЛАТЬ ТАК, ЧТОБЫ
    НЕСКОЛЬКО ВЕРХНИХ СТРОК НЕ
    СКРОЛЛИРОВАЛИСЬ ПРИ ПЕЧАТИ НА
    ЭКРАНЕ ДЛИННЫХ СООБЩЕНИЙ. СКОЛЬ-
    КО СТРОК ОСТАВИТЬ НЕПОДВИЖНЫМИ,
    РЕШИТЕ САМИ."
273 PRINT '"ЕСЛИ ТАКОЙ ЭФФЕКТ НЕ НУЖЕН - 0"
277 INPUT SCRO
279 IF SCRO>15 THEN PRINT INK 2; AT 18,7; "ЭТО СЛИШКОМ МНОГО!": ВЕР .5,20: PAUSE 50: GO TO
    265
280 LET SCRO=23-SCRO
285 GO SUB 40
290 PRINT AT 8,0;"В КАКОЙ ЛОКАЦИИ НАЧИНАЕТСЯ ИГРА?"
295 INPUT BEG
330 POKE 64821,DAV-256*INT (DAV/256): POKE 64822,INT (DAV/256)
335 POKE 64860,DAN-256*INT (DAN/256): POKE 64861,INT (DAN/256)
340 POKE 64903,MCOB: POKE 64924,(MOB-MCOB): POKE 64934, MOB: POKE 54944,MOB
345 POKE 64960,(MOB+1): POKE 64972,MOB: POKE 64997,(MOB+1)
350 POKE 64982,MOB: POKE 65012,MOB: POKE 64453,SCRO
355 POKE 65072, BEG
360 POKE 64750,VL: POKE 64772,VL: POKE 64827,VL: POKE 64810,NL:POKE 64866,NL
365 GO SUB 40
370 PRINT AT 6,0; "
    ЕСЛИ ВСЕ ДАННЫЕ ВЫ ВВЕЛИ КАК ПО-
    ЛОЖЕНО, НАЖМИТЕ <ENTER> И НАЧ-
    НЕТСЯ ГЕНЕРАЦИЯ СИСТЕМНОГО КОДА."
372 PRINT INK 2;AT 13,4; "ИНАЧЕ ВВЕДИТЕ <S> - STOP"
373 INPUT LINE A$
374 IF A$="S" THEN GO TO 9999
500 GO SUB 40
502 PRINT INK 2; BRIGHT 1; FLASH 1;AT 8,12; "ЖДИТЕ...";AT 10,5; " ИДЕТ ГЕНЕРАЦИЯ ДАННЫХ."
505 RESTORE 1000
506 DIM V$(V,VL)

```

```

508 FOR Y=1 TO V
510 READ V$(Y)
515 FOR X=1 TO VL
520 POKE (((DAV-VPP)+(Y*VP)+X)), CODE V$(Y,X)
525 IF CODE V$(Y,X)=32 THEN POKE (((DAV-VPP)+(Y*VP)+X)), 0
530 NEXT X
535 READ Z
540 POKE ((DAV-1)+(Y*VP)), Z
545 NEXT Y
550 POKE (DAV+(V*VP)), 199
600 DIM O$(MOB), L)
602 RESTORE 2000
603 DIM N$(MOB+NON+21), NL)
604 IF MCOB=0 THEN GO TO 670
605 FOR Y=1 TO MCOB
610 READ N$(Y)
615 FOR X=1 TO ML
620 POKE (((DAN-MPP)+(Y*NP)+X)), CODE N$(Y,X)
625 IF CODE N$(Y,X)=32 THEN POKE (((DAN-MPP)+(Y*NP)+X)), 0
630 NEXT X
635 READ Z
640 POKE ((DAN-1)+(Y*NP)), Z
645 READ B$
650 LET O$(Y)=B$
655 READ B$
660 LET O$(Y+MOB-MCOB)=B$
661 NEXT Y
662 FOR X=0 TO ((MCOB*NP)+1)
663 POKE (DAN+X)+((MOB-MCOB)*NP), PEEK (DAN+X)
665 NEXT X
670 FOR Y=(MCOB+1) TO (MOB-MCOB)
675 READ N$(Y)
680 FOR X=1 TO NL
685 POKE (((DAN-NPP)+(Y*NP)+X)), CODE N$(Y,X)
690 IF CODE N$(Y,X)=32 THEN POKE (((DAN-NPP)+(Y*NP)+X)), 0
695 NEXT X
700 READ Z
705 POKE ((DAN-1)+(Y*NP)), Z
710 READ B$
715 LET O$(Y)=B$
720 NEXT Y
730 FOR Y=(MOB+1) TO (MOB+NON+21)
735 READ N$(Y)
740 FOR X=1 TO NL
745 POKE (((DAN-NPP)+(Y*NP)+X)), CODE N$(Y,X)
750 IF CODE N$(Y,X)=32 THEN POKE (((DAN-NPP)+(Y*NP)+X)), 0
755 NEXT X
760 READ Z
765 POKE ((DAN-1)+(Y*NP)), Z
770 NEXT Y
775 POKE (DAN+(N*NP)), 199
780 RESTORE 3000
785 LET DMOV=DAN-(NOR*7)-5
787 LET DMOVL=DMOV-256*INT (DMOV/256): LET DMOV=INT(DMOV/256)
788 POKE 65009, DMOVL: POKE 65010, DMOV
790 FOR Y=DMOV TO (DMOV+(NOR*7)-1)
795 READ A: POKE Y, A: NEXT Y
800 RESTORE 4000
801 FOR X=65078 TO (65078+((MOB-1)*4)) STEP 4
802 POKE X, 35: POKE X+1, 62: POKE X+3, 119
803 READ A: POKE X+2, A
804 NEXT X
805 POKE 65078+(MOB*4), 201
810 LET PBS=65078+(MOB*4)+5
820 LET PBSL=FBFS-256*INT (PBS/256): LET PBSM=INT (PBS/256)

```

```

825 POKE 64931,PBSL: POKE 64932,PBSM
826 POKE 64941,PBSL: POKE 64942,PBSM
827 POKE 64969,PBSL: POKE 64970,PBSM
828 POKE 64979,PBSL: POKE 64980,PBSM
860 FOR X=1 TO 11: BEEP .1,X: NEXT X
865 FOR X=10 TO 1 STEP -1: BEEP .1,X: NEXT X
867 REM USE BY BASIC =====
868 LET SAVE=DMOV-10: LET LEN=65368-SAVE
869 LET SAVED=SAVE-256*INT (SAVE/256): LET SAVEDM=INT (SAVE/256)
870 LET LENL=LEN-256*INT (LEN/256): LET LENM=INT (LEN/256)
871 RESTORE 874
872 FOR X=0 TO 9
873 READ Z: POKE (64116+X),Z: NEXT X
874 DATA SAVED, SAVEDM, MAX, MOV, LENL, LENM, PBSL, PBSM, DMOVL, DMOV
875 GO SUB 40
885 PRINT AT 8,0;"
      ПОДГОТОВЬТЕСЬ К ВЫГРУЗКЕ НА
      ЛЕНТУ. НАЖМИТЕ <ENTER>, КОГДА
      БУДЕТЕ ГОТОВЫ."
886 PRINT INK 2; AT 12,0;"
      ЕСЛИ НЕ УВЕРЕНЫ, ЧТО ВСЕ ПРА-
      ВИЛЬНО, ТО ВВЕДИТЕ <S> - STOP."
887 INPUT A$
890 IF A$="S" THEN GO TO 9999
900 GO SUB 40
910 PRINT AT 8,11;"ЗАПИСЬ..."
915 SAVE "SYSTEM"CODE SAVE,LEN
920 SAVE "Objects" DATA O$( )
925 PRINT AT 8,0;"
      ПРЕЖДЕ, ЧЕМ БЕЙСИК-ПРОГРАММА
      И КОДЫ БУДУТ ЗАГРУЖЕНЫ В ПАМЯТЬ,
      ДОЛЖНО БЫТЬ ВЫПОЛНЕНО:" INK 2:TAB 10;"CLEAR ";SAVE-1
927 BEEP .5,20: PAUSE 50
930 PRINT PAPER 2; INK 7; BRIGHT 1;AT 15,0;"
      ОТМОТАЙТЕ ЛЕНТУ
      И ВЫПОЛНИТЕ ВЕРИФИКАЦИЮ
932 BEEP .5,20: PAUSE 50
935 PRINT AT 18,0;"
      НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КОГДА БУДЕТЕ ГОТОВЫ."
940 PAUSE 0
945 GO SUB 40
950 PRINT AT 8,9; "ВЕРИФИКАЦИЯ..."
955 VERIFY "system"CODE SAVE,LEN
960 VERIFY "objects" DATA O$( )
965 CLS
970 GO TO 9999
999 REM ГЛАГОЛЫ DATA 0-8 =====
1000 DATA "ИДИТИ",0,"СЕВЕР",0,"С",0,"ЮГ",0,"Ю",0,"ВОСТОК",0,"В",0,"НАПРАВО",0,"ЗАПАД",0,"З",
      0,"НАЛЕВО",0,"ВВЕРХ",0,"Х",0,"ВНИЗ",0,"Н",0,"ВЗЯТЬ",1,"БРАТЬ",1,"ПОЛОЖИТЬ",2,
      "БРОСИТЬ",2,"ПРОВЕРИТЬ",3,"ОСМОТРЕТЬ",3,"СМОТРЕТЬ",4,"ИНВЕНТАРЬ",5,"И",5,"СТОП",6,
      "КОНЕЦ",6,"СОХРАНИТЬ",7,"ЗАГРУЗИТЬ",8
1002 REM ДАННЫЕ ДЛЯ ПРОЧИХ ГЛАГОЛОВ (КОМАНД) =====
1999 REM ОБЪЕКТЫ, КОТОРЫЕ ИЗМЕНЯЮТ СВОЕ ОПИСАНИЕ В ХОДЕ ИГРЫ==
2019 REM ПРОЧИЕ ОБЪЕКТЫ =====
2189 REM ПРОЧИЕ СЛОВА. РАСПОЗНАВАЕМЫЕ ПРОГРАММОЙ =====
2190 DATA "СЕВЕР",MOV+1,"С",MOV+1,"ЮГ",MOV+2,"Ю",MOV+2,"ВОСТОК",MOV+3,"В",MOV+3,"НАПРАВО",
      MOV+3,"ЗАПАД",MOV+4,"З",MOV+4,"НАЛЕВО",MOV+4,"ВВЕРХ",MOV+5,"Х",MOV+5,"ВНИЗ",MOV+6,
      "Н",MOV+6,"СМОТРЕТЬ",99,"ИНВЕНТАРЬ",99,"И",99,"СТОП",99,"КОНЕЦ",99,"СОХРАНИТЬ",99,
      "ЗАГРУЗИТЬ",99
2999 REM ДАННЫЕ О ВОЗМОЖНЫХ ПЕРЕМЕЩЕНИЯХ =====
3999 REM ИСХОДНОЕ РАЗМЕЩЕНИЕ ОБЪЕКТОВ =====
9990 GO SUB 40: SAVE "ABS" LINE 2
9991 REM
9993 PRINT AT 8,0;"ОТМОТАЙТЕ ЛЕНТУ ДЛЯ ВЕРИФИКАЦИИ."
9994 BEEP .1,20: PAUSE 50

```

```

9995 PRINT AT 12,0;"НАЖМИТЕ ЛЮБУЮ КЛАВИШУ, КОГДА БУДЕТЕ ГОТОВЫ.": PAUSE 0
9996 GO SUB 40
9997 PRINT AT 8,7; "ВЕРИФИКАЦИЯ....."
9998 VERIFY "ABS"
9999 BORDER 7: PAPER 7: INK 0

```

Теперь дополните текст Мастер-копии строками данных для демонстрационной программы из Листинга_2 или наберите их отдельно (и сохраните на ленте). В последней случае загрузите мастер-копию, объедините ее при помощи MERGE с данными для демонстрационной игры из Листинга_2 и сохраните полученную программу командой RUN 9990. Эту выгрузку надо делать на рабочую ленту (назовем ее лента_A).

Листинг_2. Данные для демонстрационной игры. (Объединить с файлом "ABS")

```

1003 DATA "ОТКРЫТЬ",9,"ОТПЕРЕТЬ",10,"КОПАТЬ",11,"ВЫКОПАТЬ",11,"ЗАЖЕЧЬ",12,"ГАСИТЬ",13,
"ПОГАСИТЬ",13,"ПОДНЯТЬСЯ",14,"ЗАЛЕЗТЬ",14,"ЛЕЗТЬ",14,"СПУСТИТЬСЯ",15,"СЛЕЗТЬ",15
1005 DATA "ЗАКРЫТЬ",16,"ШВЫРНУТЬ",17,"ЗАКОПАТЬ",18,"ЗАПЕРЕТЬ",19,"ОДЕТЬ",20,"НАДЕТЬ",20,
"СНЯТЬ",21
1007 DATA "ПРЫГАТЬ",22,"СПРЫГНУТЬ",22,"ЗАПРЫГНУТЬ",22,"ИСПОЛЬЗОВАТЬ",23
2000 DATA "ПЛЕД",1,"ШЕРСТЯНОЙ ПЛЕД","ПЛЕД, НАДЕТЫЙ НА ПЛЕЧИ"
2001 DATA "ФАКЕЛ",2,"МАЛЕНЬКИЙ ФАКЕЛ","ГОРЯЩИЙ ФАКЕЛ"
2020 DATA "ШКАТУЛКА",3,"ДЕРЕВЯННАЯ ШКАТУЛКА"
2021 DATA "ЛОПАТА",4,"КОРОТКАЯ ЛОПАТА"
2022 DATA "КЛЮЧ",5,"БРОНЗОВЫЙ КЛЮЧ"
2023 DATA "ЩЕПКИ",6,"ДЕРЕВЯННЫЕ ЩЕПКИ"
2024 DATA "АЛМАЗ",7,"КРУПНЫЙ АЛМАЗ"
2191 DATA "ДВЕРЬ",16,"БУФЕТ",17,"ЛУЖАЙКА",18,"СТУПЕНИ",19,"ЛЕСТНИЦА",20
2192 DATA "ВОРОТА",21,"КОПАТЬ",22,"ЯМУ",22,"САД",23,"ПОЛЕ",24
2194 DATA "СЕЙФ",25,"ПЯТНО",26," ПРЫГАТЬ",27
3001 DATA 1,0,3,0,0,0,0
3002 DATA 2,0,0,0,0,0,6
3003 DATA 3,1,7,0,0,0,0
3004 DATA 4,0,9,5,0,0,0
3005 DATA 5,0,10,0,4,0,0
3006 DATA 6,0,0,7,0,2,11
3007 DATA 7,3,13,8,6,0,0
3008 DATA 8,0,0,9,7,0,0
3009 DATA 9,4,0,10,8,0,0
3010 DATA 10,5,15,0,9,0,0
3011 DATA 11,0,0,0,0,6,0
3012 DATA 12,0,0,13,0,0,0
3013 DATA 13,7,0,14,12,0,0
3014 DATA 14,0,0,15,13,0,0
3015 DATA 15,10,0,0,14,0,0
4000 DATA 4,10,0,1,0,0,0,0,0

```

Теперь наберите загрузчик будущей демонстрационной игры - это Листинг_3 и сохраните его командой RUN 200. Эту выгрузку делаем на новую ленту (Ленту_B). Ленту назад не отматывайте.

Листинг_3. Загрузчик демонстрационной игры. Файл "INTRO".

```

10 CLEAR 49999: LOAD "chr"CODE 50000
20 BORDER 7: PAPER 7: INK 0: CLS : POKE 23606,80: POKE 23607,194
30 PRINT INK 1; BRIGHT 1; AT 2,6;"ДЕМОНСТРАЦИОННАЯ ИГРА"
40 PRINT AT 4,11;"СОСТАВЛЕНА"" " " С ИСПОЛЬЗОВАНИЕМ ПРОГРАММЫ"
50 PRINT PAPER 0; INK 6; BRIGHT 1; AT 7,3;" ADVENTURE BUILDER SYSTEM "
60 PRINT AT 9,5; "T.D. FROST + ""ИНФОРКОМ""""
70 PRINT PAPER 0; INK 5; BRIGHT 1; AT 12,9;" ЗАГРУЗКА... "

```



```

1800 LET R$(1)="В СЛЕДУЮЩЕМ ВЫПУСКЕ ""ZX-РЕВЮ""": GO TO 50
1900 IF (PEEK F=8 OR PEEK F=9) AND NO=16 AND PEEK (F+4)=0 THEN POKE (F+4),1: LET R$(1) =
    "О.К. ТЕПЕРЬ ДВЕРЬ ОТКРЫТА.": GO TO 50
1902 IF PEEK F=14 AND NO=21 AND PEEK (F+7)=0 THEN LET R$(1)="НЕ ОТКРЫВАЮТСЯ, ЗАКРЫТЫ
    ИЗНУТРИ.": GO TO 50
1905 IF PEEK F=15 AND NO=21 AND PEEK (F+7)=0 THEN POKE (F+7),1: LET R$(1)="О.К. ТЕПЕРЬ
    ВОРОТА ОТКРЫТЫ.": GO TO 50
1910 IF (((PEEK F=8 OR PEEK F=9) AND PEEK (F+4)=1 AND NO=16) OR (((PEEK F=14 OR PEEK F=15)
    AND PEEK (F+7)=1) AND NO=21)) THEN LET R$(1)="ДА ВЕДЬ И ТАК УЖЕ ОТКРЫТО!": GO TO 50
3315 LET r$(1)="В СЛЕДУЮЩЕМ ВЫПУСКЕ ""ZX-РЕВЮ""": GO TO 50
7010 PRINT T$
7011 PRINT "
    АВЫ НАХОДИТЕСЬ В ЧИСТОМ И АККУ-Г
    АРАТНОМ АМБАРЕ. ФЕРМЕР, КОТОРО-Г
    АМУ ОН ПРИНАДЛЕЖИТ, ПО-ВИДИМОМУГ
    АЛЮБИТ СВОЕ ДЕЛО. G"
7012 PRINT B$
7019 RETURN
7020 PRINT T$
7021 PRINT "
    АЗАБРАВШИСЬ ПО ЛЕСТНИЦЕ НАВЕРХ,Г
    АВЫ ПОПАЛИ КАК БЫ НА ОТКРЫТЫЙГ
    АЧЕРДАК. СКОРЕЕ ЕГО МОЖНО ОПИ-Г
    АСАТЬ, КАК БОЛЬШОЕ ХРАНИЛИЩЕГ
    АНАД КОМНАТОЙ. G"
7022 PRINT B$
7029 RETURN
7030 PRINT T$
7031 PRINT "
    АВЫ ВЫШЛИ НА ДВОР ФЕРМЫ. НИКА-Г
    АКИХ ЖИВОТНЫХ НЕ ВИДНО. НА СЕ-Г
    АВЕРЕ НАХОДИТСЯ АМБАР, НА ЮГЕ -Г
    А-ПОЛЕ. G"
7032 PRINT B$
7039 RETURN
7040 PRINT T$
7041 PRINT "
    АУЮТНАЯ ГОСТИННАЯ КРЕСТЬЯНСКОГОГ
    АДОМА. ВЫ ЧУВСТВУЕТЕ НА СЕБЕГ
    АЧЕЙ-ТО ВЗГЛЯД. G"
7042 PRINT B$
7049 RETURN
7050 PRINT T$
7051 PRINT "
    АКАЖЕТСЯ, ЧТО ЭТОЙ СТОЛОВОЙГ
    АКОМНАТОЙ ДАВНО НИКТО НЕ ПОЛЬ-Г
    АЗОВАЛСЯ. НАВЕРНОЕ, ЗДЕСЬ НИКТОГ
    АНЕ ЖИВЕТ. G"
7052 PRINT B$
7059 RETURN
7060 PRINT T$
7061 PRINT "
    АКОНЮШНЯ СОВЕРШЕННО ПУСТА. НЕТГ
    АДАЖЕ ХАРАКТЕРНОГО ЗАПАХА ЖИ-Г
    АВОТНЫХ. ЛЕСТНИЦА ВЕДЕТ НА ЧЕР-Г
    АДАК, ВИДНЫ ТАКЖЕ СТУПЕНИ,Г
    АСКРЫВАЮЩИЕСЯ ГДЕ-ТО ВНИЗУ, ВГ
    АТЕМНОТЕ. G"
7062 PRINT B$
7069 RETURN
7070 PRINT T$
7071 PRINT "
    АВЫ ВЫШЛИ В ОТКРЫТОЕ ПОЛЕ. ОНОГ
    АБЫЛО НЕДАВНО ВСПАХАНО, НО НИ-Г

```

АЧЕГО ПОСАЖЕНО НЕ БЫЛО. МОЖНО
 АИДИ В ЛЮБОМ НАПРАВЛЕНИИ. G“

7072 PRINT B\$
 7079 RETURN
 7080 PRINT T\$
 7081 PRINT “

АВЫ НАХОДИТЕСЬ У ДВЕРИ ФЕРМЕР-Г
 АСКОГО ДОМА. ДОМ СДЕЛАН ОЧЕНЬГ
 АДОБРОТНО И КРЕПКО, КАК БУДТОГ
 АХОЗЯЕВАМ БЫЛО ОТ ЧЕГО СКРЫ-Г
 АВАТЬСЯ. G“

7082 PRINT B\$
 7089 RETURN
 7090 PRINT T\$
 7091 PRINT “

АПЕРЕД ВАМИ НЕБОЛЬШАЯ ПРИХОЖАЯ. G
 АВ УГЛУ СТОИТ БУФЕТ И БОЛЬШЕГ
 АНИЧЕГО ЗДЕСЬ НЕТ. НА ВОСТОКЕГ
 АКУХНЯ, НА СЕВЕРЕ - ГОСТИННАЯ. G”

7092 PRINT B\$
 7099 RETURN
 7100 PRINT T\$
 7101 PRINT “

АКУХНЯ СВЕРКАЕТ БЕЛИЗНОЙ И ЧИ-Г
 АСТОТОЙ. В ВОЗДУХЕ ОПРЕДЕЛЕННОГ
 АЧУВСТВУЕТСЯ ЗАПАХ ЧЕГО-ТО ПРИ-Г
 АЯТНОГО. G”

7108 PRINT B\$
 7109 RETURN
 7110 IF PEEK (0+9)<>99 THEN PRINT TAB 5; PAPER 7; INK 0; “ЗДЕСЬ НИЧЕГО НЕ ВИДНО! ”: RETURN
 7111 PRINT T\$
 7112 PRINT “

АВ ОТЛИЧИЕ ОТ БОЛЬШИНСТВА ОБЫЧ-Г
 АНЫХ ПОГРЕБОВ, ЭТОТ - ЯВНО НЕГ
 АИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ ВСЯ-Г
 АКИХ ЗАПАСОВ. А В УГЛУ МОЖНОГ
 АРАЗГЛЯДЕТЬ БОЛЬШОЙ СТАРОМОДНЫЙГ
 АСЕЙФ СО ВСТРОЕННЫМ В НЕГОГ
 АБРОНЗОВЫМ ЗАМКОМ. G”

7113 PRINT B\$
 7119 RETURN
 7120 PRINT T\$
 7121 PRINT “

АЭТО ТУПИК. МОЖНО БЫЛО СЮДА ВО-Г
 АОБЩЕ НЕ ПРИХОДИТЬ. G”

7122 PRINT B\$
 7129 RETURN
 7130 PRINT T\$
 7131 PRINT “

АЭТА ОЧЕНЬ УЗКАЯ ДОРОГА ВЕДЕТ СГ
 АВОСТОКА НА ЗАПАД. НО, КАЖЕТСЯ, G
 АМОЖНО ПРОЙТИ И НА СЕВЕР. G”

7132 PRINT B\$
 7139 RETURN
 7140 PRINT T\$
 7141 PRINT “

АДорога УПИРАЕТСЯ В ВОРОТА, ЗАГ
 АНИМИ РАСПОЛОЖЕН ПРИУСАДЕБНЫЙГ
 АСАД. G“

7142 PRINT B\$
 7149 RETURN
 7150 PRINT T\$
 7151 PRINT “

АЭТО ДОВОЛЬНО ПУСТЫННЫЙ САД, G
 АКоторый СКОРЕЕ ПОХОЖ НА ЛУЖАЙ-Г

наборе.

Головная программа ABS.

После старта головная программа ABS выдает серию экранных запросов, по которым надо ввести необходимые данные. Проследите по демонстрации, как и в каком формате вводятся данные по этим запросам, а мы сейчас их подробно рассмотрим. После каждого заголовка раздела стоит число в скобках, обозначающее, что должно вводиться на этот запрос в демонстрации.

1. Количество прочих глаголов. (23). Эти глаголы заносятся в строки DATA 1003...1999 (Листинг_2) Например:

```
DATA "ОТКРЫТЬ", 9, "ОТПЕРЕТЬ", 10
```

Формат: глагол (полностью) + присвоенный ему номер. Поскольку номера с 0...по 8 уже присвоены стандартным командам, то первая пользовательская команда (глагол) начинается с номера 9. Слова синонимы, такие как ГАСИТЬ, ПОГАСИТЬ, имевшие одинаковое действие, но разное написание, вводятся порознь, но им присваивается один и тот же номер. Тем не менее, в ответ на запрос "Количество глаголов?" их надо считать как отдельные слова.

В системе ABS применяется следующий формат ввода команды: первое слово считается глаголом, последнее - существительным или направлением (в случае перемещения). Можно в отдельных случаях ввод осуществлять и одним словом (буквой), но для реализации этого надо, чтобы это слово (буква) было занесено и в список глаголов и в список существительных (мы еще вернемся к этому моменту позднее).

Основные глаголы заданы в строке 1000 Мастер-копии программы ABS (Листинг_1). Основные существительные - в строке 2190 этого же листинга.

Такие действия, как ИДТИ НА ВОСТОК, можно вводить также и ИДТИ НА В, или ИДТИ В, или ВОСТОК, или только буквой В, а также словами ИДТИ НАПРАВО или просто НАПРАВО. При помощи одного слова можно также проверить ИНВЕНТАРЬ (или просто И), ввести СТОП и другие - те, которые одновременно заданы и в строке 1000 и в строке 2190. А вот глаголы взять или ПОЛОЖИТЬ не могут использоваться отдельно, без существительного и, поэтому, они не заданы в строке 2190.

На тот случай, если Вы захотите изменить основные глаголы, заданные в Листинге_1, в строке 1000, обращаем Ваше внимание на то, что общее количество глаголов в строке 1000 отражено также в строке 126 этого листинга. Это число - 28. Если будете изменять данные в строке 1000, проконтролируйте и число в строке 126. Аналогично и для строки 2190 - полное число слов в этой строке отражено числовыми параметрами в строках 603 и 730. Это число - 21.

2. Символы идентификации команды (4). Для того, чтобы пользователю не приходилось полностью набирать длинные слова (и не дай Бог при этой сделать ошибку), программа может идентифицировать команду не по всему слову, а по нескольким первым символам. Например, вместо команды прыгать можно дать команду ПРЫГ (если установлено число 4) или ПРЫ (если установлено 3). Это число может быть задано в диапазоне от 3 до 7. Обычное число - 4 или 5. Для детей - 4, для взрослых программы посложнее, можно задать 5, так как при большом числе слов возможны ситуации, когда первые 4 буквы разных слов совпадают. Для числа 5 - это происходит значительно реже. Число 3 еще как-то приемлемо для несложных английских программ, но в русском языке существуют приставки, которые могут быть одинаковыми у разных слов и очень возможны ошибки.

3. Количество объектов (9). Это количество объектов в программе, которые могут быть инвентарем, т.е. которые пользователь может брать, переносить с места на место или как-то использовать. Подробности, как заносить список этих объектов в листинги, смотрите в комментариях к пункту 6.

4. Символа идентификации объекта (4). Это то же, что и п.2, но для второй части команды - для существительных или направления. Эти числа могут и различаться, но лучше, чтобы сокращения для глаголов и существительных делались по единым правилам, пользователь к ним проще привыкает и чаще пользуется.

5. Максимальная длина описания объекта (22). Это количество символов, которое может занимать самое длинное описание объекта (например, ПЛЕД, НАДЕТЫЙ НА ПЛЕЧИ).

6. Количество объектов, способных изменять свое описание по ходу игры (2). Этот пункт имеет отношение к объектам типа "ФАКЕЛ" - "ГОРЯЩИЙ ФАКЕЛ" и т.п. Данные на эти объекты заносятся в строки 2000 ... 2018 (см. Листинг_2 демонстрационной программы), например:

```
DATA "ПЛЕД", 1, "ШЕРСТЯНОЙ ПЛЕД", "ПЛЕД, НАДЕТЫЙ НА ПЛЕЧИ"  
DATA "ФАКЕЛ", 2, "МАЛЕНЬКИЙ ФАКЕЛ", "ГОРЯЩИЙ ФАКЕЛ"
```

Это были объекты 1 и 2, а их измененные версии будут потом названы объектами 8 и 9. Поэтому при ответе на запрос о количестве предметов, являющихся инвентарем, надо задавать число 9, хотя в строках 2000-2024 заданы только 7 предметов. Кроме строк 2000 и 2001 (объекты, которые могут изменяться), остальной инвентарь, который не может изменять свое состояние, задан в строках 2020-2188 в формате:

```
DATA "ШКАТУЛКА", 3, "ДЕРЕВЯННАЯ ШКАТУЛКА"
```

7. Другие слова, распознаваемые программой (13). Эти слова заданы в строках с 2191. Первый присваиваемый им номер рассчитывается по формуле:

(количество объектов) + 7

Поэтому в демонстрационной программе первое слово дверь имеет номер 16.

8. Максимальное число объектов, которые одновременно можно взять и иметь при себе (2).

9. Общее число локаций (15).

10. Фактор скроллинга (0). Если Вы хотите, чтобы определенное количество верхних строк не скроллировалось при вводе команд, то можете до запросу задать это число. В демонстрационном примере это не используется, поэтому введен 0.

11. Стартовая локация (7). Это номер локации, в которой начинается игра.

Возможные перемещения.

Со строки 3000 (до строки 3999) в Листинге_2 находится таблица расположения локаций, она же определяет и возможность взаимных перемещений между локациями. Первое число в строке DATA - это номер локации. Всего их 15 (в данной демонстрационной программе). Их расположение и наименование для демонстрационной программы показано на плане (рис. 1).

Формат задания допустимых перемещений следующий. Первая цифра в строке DATA - это номер локации. Далее идут номера других локаций, в которые можно попасть из данной, если идти в разных направлениях. Всего 6 направлений, по-порядку соответственно СЕВЕР, ЮГ, ВОСТОК, ЗАПАД, ВВЕРХ и ВНИЗ. Так, например, для 1 локации возможен только переход в 3 локацию при движении на ЮГ:

```
DATA 1, 0, 3, 0, 0, 0, 0
```

а для 7 локации - допустимы 4 направления:

```
DATA 7, 3, 13, 8, 6, 0, 0
```

в 3 локацию, если идти на СЕВЕР,

в 13 -//- -//- ЮГ.

в 8 -//- -//- ВОСТОК.

в 6 -//- -//- ЗАПАД

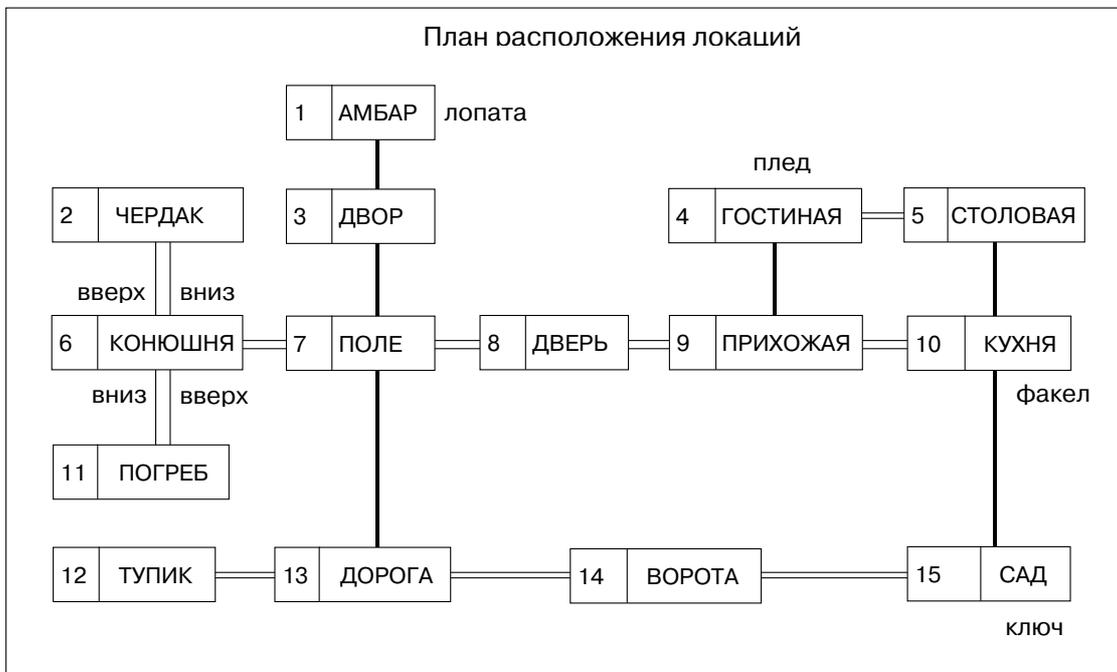


Рис. 1

Расположение объектов.

И, наконец, в строке 4000 задается исходное расположение объектов, являющихся инвентарем. Формат записи такой. По порядку, начиная с объекта номер 1 (ПЛЕД), задаются номера локаций, в которых этот предмет находится. Если предмет в данный момент не виден, то заносится 0. Всего в строке DATA будет столько цифр, сколько задано объектов, являющихся инвентарем (для нашей демонстрационной программы - 9).

DATA 4, 10, 0, 1, 0, 0, 0, 0, 0

Это означает, что ПЛЕД расположен в 4 локации, ФАКЕЛ - в 10 локации, ШКАТУЛКА - пока не видна и т.д., для всех 9 предметов, являющихся инвентарем (включая и те, которые изменяют свое состояние в процессе игры). Если Вы захотите, чтобы вначале игры какой-нибудь объект находился у Вас в руках, то надо для него задать номер 99 - это будет соответствовать "объект взят".

Загрузчик.

Листинг_3 является загрузчиком. Его функции - переустановка RAMTOP, загрузка русского символического набора и его активизация, а также загрузка основной Бейсик-программы демонстрационной игры (Листинг_4).

Демонстрационная программа.

Автостарт программы происходит со строки 9940, где после загрузки кодового блока и блока данных, начинается задание параметров, сохраненных в кодовом блоке, задаются массивы для переменных и т.д. Затем задаются символы UDG-графики А-Н. Последние служат для изображения рамок, оконтуривающих текст. Они встречаются в тексте программы в кавычках. Весь текст за редким исключением (символы "Y" или "N") набирается по-русски, а встречающиеся символы с "А" по "Н" (например, в строке 9988, строки с 7000 и др.) - это символы UDG-графики, изображающие рамки.

Подготовка исходных данных заканчивается строкой 9989, где управление передается на начало исполнения программы - строку 100. В процессе отладки, когда Вы будете останавливать программу, вновь стартовать ее можно, подав команду GO TO 100, но не RUN 100.

(Окончание - в следующем номере)

Компьютерная новелла

(С) Матвеев Ю.А., 1993 г

СТРАТЕГИЯ КАПИТАНА КРЕНОНА

(по мотивам программы REBEL STAR)

Часть 2. Столкновение

Они смотрели на белый, ослепительно яркий диск Венеры, который занимал добрую половину обзорного экрана. У самой кромки диска, в фиолетовой дымке, гигантским алмазом сверкал в солнечных лучах орбитальный комплекс "Венера-Центр", окруженный целой флотилией грузовых и пассажирских кораблей разнообразных типов и марок.

Полет близился к концу. Через час космический лайнер "Викинг" должен был достичь границы действия гравитационного поля комплекса. Кренон сидел в пол-оборота к экрану и часто поглядывал на внука, наблюдая за его реакцией на великолепный космический пейзаж, созданный природой и человеком.

- Здорово! - не отрывая глаз от экрана с восхищением сказал юноша. - Это не то, что на наших тренажерах в школе.

Кренон похлопал внука по плечу:

- Ничего, учишь пока, скоро сам летать будешь.

- Скорей бы, - вздохнул будущий пилот сверхскоростных звездных кораблей. Он развернулся в кресле и хитро прищурился. - Дед, мы уже скоро прилетим, а ты еще не рассказал мне про вашу встречу с инопланетянами.

Кренон посмотрел на часы. Через два часа они будут на Венере, в гостях у его друга детства, Дона Кейлони, и тогда ему трудно будет найти свободную минуту, чтобы спокойно рассказать внуку о той трагической встрече с коварными звездными монстрами.

Это случилось во времена великих завоеваний человечества. Расширялись границы обитания, открывались новые миры, строились огромные звездные лайнеры, способные переносить все большее количество землян на дальние космические колонии. Человеческая цивилизация заняла достойное место во Вселенной и была одной из самых молодых и быстрорастущих. Космическое пространство на десятки парсеков от Солнца было досконально изучено и тщательно исследовано, белых пятен оставалось все меньше, однако...

* * *

Эта первобытная планета ему сразу не понравилась. Вполне подходящая для жизни, она по каким-то не вполне понятным причинам, еще долгое время оставалась неизученной. Странно, что Земля, испытывающая постоянную нехватку пригодных для жизни планет, так вяло проводила исследования на Зеро. В название планеты первооткрыватели вложили двойной смысл: с одной стороны этот мир некоторое время действительно был перевалочной базой для крупных межзвездных экспедиций, своего рода "точкой отсчета" новых завоеваний. С другой стороны, не внушающие оптимизма скудная растительность и серенький животный мир, еле умещавшиеся на единственном материке, который скорее походил на остров, омываемый со всех сторон холодным океаном. Планета не оправдала надежд космобиологов и серьезные исследователи с разочарованием умывали руки.

Один единственный город, построенный колонистами, располагался на юге материка. Там же находился и космопорт. Север, из-за своей труднодоступности, был малоизучен и здесь оставались обширные земли, где еще не ступала нога человека.

Периодически из числа колонистов формировались экспедиции для изучения природы материка, которые с каждым разом уходили все дальше и дальше на Север. Жизнь на Зеро текла своим чередом и, казалось, ничто не могло поколебать ее спокойствие и размеренность. Однако вскоре произошло непредвиденное.

В один ясный, холодный день прервалась связь с экспедицией профессора Джонсона, который был ведущим специалистом по флоре и фауне дикого Севера планеты. Спасательный отряд, собранный наспех из добровольцев и в срочном порядке отправленный на поиски, бесследно исчез на следующие сутки. Тогда руководство колонии и обратилось за помощью в Зональный Центр Космической Безопасности.

Капитан Кренон возглавил десантный отряд из шестнадцати человек. В его подчинении были неплохо подготовленные десантники. Многие из них уже прошли хорошую закалилку в поисковых и спасательных операциях.

Получив задание, отряд капитана Кренона немедленно приступил к его завершению. Город выделил десантникам легкий винтолет, на котором сразу из космопорта их доставили в район поиска пропавшей экспедиции.

Надо сказать, что северная часть материка буквально утопала в смердящих болотах, а небольшие участки суши покрывали непроходимые заросли ползучих лиан и болотного тростника. Иногда здесь можно было встретить не очень высокие пальмообразные деревья с толстыми сочными листьями ядовито-фиолетового цвета. Живности здесь было немного, если не считать болотных крыс размером с корову, да местных крокодилов с длинными электрическими щупальцами. И с теми и с другими встреча не сулила ничего хорошего: острыми, словно лезвие бритвы, клыками, болотные крысы могли, пожалуй, перегрызть стальной канат в руку толщиной, а об электрических крокодилах, которых колонисты называли свамперами, и говорить не приходилось.

Отряд шел цепью. Вот уже вторые сутки, используя светосабли, они продирались сквозь заросли лиан. От сырого воздуха, наполненного зловониями болотных испарений, мутило, а ноги деревенели от постоянного сопротивления вязкой хлюпающей жижи.

В два часа пополудни Кренон связался с Центром и доложил о результатах первых суток поисков. Получив необходимые инструкции он решил дать отдохнуть уставшим десантникам.

Отряд расположился на небольшой поляне, окруженной со всех сторон густым кустарником. На привале они обменялись мнениями. Рыжеволосый Зира Кама скептически относился к приказу центра продолжить поиски. Он протирал забрызганное грязью лазерное ружье и угрюмо бубнил что-то про свамперов и крыс. В той или иной степени его поддерживали еще трое: Джекки Доннелли, Лазарь Шарки и дон Кейлони.

- Им там легко рассуждать, - ворчал Шарки. - Пойдите туда, пойдите сюда... Пусть сами подышат местным ароматом. Этого профессора наверняка давно сожрали свамперы, а мы тут ищем сами не знаем что.

- Да ладно тебе, - одернул его Дон Кейлони, уловив мрачный взгляд Кренона. - Скоро вернемся, не вечно же нам здесь блуждать. Правда, капитан?

Кренон многозначительно посмотрел на Дона и, расстегивая крючки комбинезона, повернулся к Диане Вудлей - врачу отряда:

- Больные есть?

- Все здоровы, - ответила Диана.

- Тогда какого дьявола мне здесь закатывают истерику? - вскипел капитан, поднимаясь с надувного походного кресла, давая тем самым понять, что любые споры закончены. - Значит так. Здесь разбиваем постоянный лагерь. Со мной продолжат поиски десять человек. Остальные, включая врача, остаются в лагере до нашего возвращения. Вопросы?..

- Кто остается? - спросил Люк Соло.

Что? Я разве не сказал? - Кренон задумался. - В лагере останутся капрал Хикс, Дэниел Дар, Хадсон, Диана Вудлей и Лазарь Шарки. Займетесь благоустройством лагеря. Ну и ужин хороший приготовьте. Мы к вечеру вернемся.

- Смотрите только сами себя свамперам не скормите, - добавил любимец публики Хенли, разряжая обстановку.

Кренон оставлял в лагере самых молодых десантников, рассчитывая лишь на опытных партнеров.

Спустя полчаса группа из одиннадцати человек продолжила поиски. Тогда они еще не

знали, что в лагерь им не суждено будет вернуться, а некоторые из них и вовсе навсегда останутся в этих проклятых Богом болотах.

Когда в воздухе прогремели выстрелы, разрывая привычную тишину, Кренон остановился и прислушался. Шедший рядом с ним Дон Кейлони ловким движением перевел лазерное ружье в боевое положение и вопросительно посмотрел на него. Командир десантников включил рацию:

- Кто стрелял? Прием...

Какое-то время в эфире царила мертвая тишина. Все замерли в ожидании ответа, затем на связь вышел Сью Шоу. Его хриплый, вечно простуженный голос трудно было с кем-нибудь спутать:

- Это я. По-моему, мы забрели в гости к болотным крысам. Одна меня только что чуть не слопала. Вы там смотрите в оба. Не нравится мне все это.

- Слушайте мой приказ,- решительно сказал Кренон. - Держитесь ближе друг к другу. Оружие привести в боевую готовность.

Он посмотрел налево и увидел как к ним с Доном Кейлони идут Хенли и Курт Левин. Справа приблизились Люк Соло, Джим Дигриз и Эйра Кама. Кренон подал знак рукой, давая понять, что дистанция стала оптимальной, и они продолжили путь.

Вскоре им вновь пришлось преодолевать непроходимые заросли ползучих лиан. Дон обнажил светосаблю. Неожиданно справа зашелестели упругие стебли и прямо перед ними, обжигая горячим дыханием, появилась огромная серая морда болотной крысы. С черной, покрытой жесткой щетиной клыкастой пасти стекала желтая пена. Готовясь к прыжку, крыса попятилась и угрожающе приподнялась на задние когтистые лапы. Все произошло настолько быстро, что Дон даже не успел ударить хищника светосаблей. Кренон, шедший следом, отпрыгнул в сторону и два раза нажал на пусковую кнопку фотонного излучателя. После первого выстрела крыса с дымящимся брюхом по инерции еще двигалась к Дону, но когда в раскрытую пасть ударил второй луч, агрессор, заливая траву малиновой кровью, в предсмертных конвульсиях завалился на бок. Длинный, похожий на канат, облезлый хвост крысы несколько секунд судорожно метался по траве, пока не замер.

- Спасибо, друг, - Дон благодарно посмотрел на капитана.

У Кренона дрожали руки, но он старался не подавать виду:

- Идем...

Местное светило стояло почти в зените, однако тепла практически не ощущалось. Толиман - так называлась звезда, вокруг которой вращалась планета. Очень похожая на Солнце, эта звезда, однако, не могла согреть Зеро, из-за обилия космической пыли, плотным облаком прикрывавшей планету от света. Вероятно в далеком будущем из этого скопления пыли образуется спутник Зеро и тогда света станет несомненно больше и жизнь на планете начнет развиваться бурными темпами. А пока, скорее похожая на царство теней, планета оставалась холодным шариком, населенным опасными хищниками, выжившими в жестокой схватке с природой. С этими мыслями Кренон пересек небольшое болотце и вышел на обширное сухое пространство, покрытое низкой травой и мхом. То, что он увидел, никак не вписывалось в рамки его представлений об этой планете. Прямо по центру поляны, образуя правильную окружность, стояли несколько тщательно отполированных булыжников высотой с человеческий рост. В центре окружности располагался гладкий плоский камень красного цвета таких размеров, что на нем могли без особого труда уместиться пара взрослых болотных крыс. Пространство вокруг этого странного сооружения было усеяно костями. Картинка из древних фильмов про кровожадных туземцев. Кренон застыл в нерешительности. Он не заметил, как сзади тихо подошли Дон Кейлони и Курт Левин.

- Что скажешь, командир? - спросил Дон, ловким движением убирая светосаблю в ножны.

Кренон не ответил. Дон хотел еще что-то спросить, но его слова потонули в треске близких выстрелов. Стреляли сразу из нескольких стволов. Потом они услышали чьи-то выкрики и ругательства. Курт Левин, не говоря ни слова, выхватил из кобуры лазерный пистолет и помчался через всю поляну на звук выстрелов. Кренон уже успел понять, что

отряд столкнулся с целым полчищем болотных крыс. Когда все стихло на связь вышел Леон Троцкий:

- У нас все в порядке. Завалили парочку крысенушек. Никто не пострадал... Капитан, мы на берегу ручья, переходить?

- Подождите,- Кренон смотрел на Джима Дигриза, который стоял вдалеке, на противоположном конце поляны и подавал какие-то знаки. - Джим, вижу тебя. Что случилось?

- Там кто-то есть,- Дигриз показывал рукой на север, в сторону ручья. - На крыс вроде не похоже.

- Я тоже заметил какое-то движение. - Это был голос Сью Шоу, который замыкал цепь десантников. Элмер Кренон внимательно посмотрел вокруг. Странное сооружение из камней и, замеченное многими движение на другом берегу ручья, заставляли задуматься. В голове вихрем пронеслись кровавые сцены расправы загадочных аборигенов с мирной исследовательской экспедицией профессора Джонсона и с колонистами, ушедшими на ее поиски.

Удивительным было то, что ни в одном справочнике по этой планете не было ни слова о какой-либо, пусть даже примитивной, форме разумной жизни. Видимо, недаром Джонсон уделял такое пристальное внимание северной части материка. Перспектива оказаться на религиозном обряде аборигенов в качестве жертвы, приносимой в угоду местному божеству, Кренона конечно же не устраивала. Но и отступать ему совсем не хотелось. Он успокаивал себя тем, что вооруженным и хорошо обученным солдатам не составят особого труда разобраться с доисторическими туземцами, стоявшим на несомненно более низком уровне развития. Ну разве могут спорить с лазерным оружием каменные топоры?

Кренон хотел было уже отдать приказ собратиться всем на поляне как в воздухе рядом с ним со свистом пронеслась стрела и глубоко воткнулась в ствол одиноко стоящей фиолетовой пальмы в пяти шагах от него. Невидимый лучник прятался в небольшой роще ниже по ручью.

- Быстрее сюда!- Это кричал Дон Кейлони, который вовремя сориентировался и уже успел спрятаться за ближайшими кустами. Возможно, его окрик спас Кренону жизнь: вторая стрела просвистела, рассекая воздух, прямо над тем клочком земли, на котором только что стоял капитан. Эфир заполнили голоса и крики десантников. Стрелы летели с противоположного берега ручья сразу из нескольких мест. Похоже, что лучников было много, может быть десятков, а то и два.

- Командир, говорит Джекки Доннелли. У нас тут ранили Сью Шоу. Я осмотрел стрелу. Облита какой-то гадостью. Похоже на яд...

- Я в порядке,- скрипя зубами перебил напарника Сью. - Задета только кисть руки.

- Вы как хотите, а я открываю огонь,- решительно прозвучал в эфире женский голос. Это была Васкез - молодая горячая девушка, только недавно закончившая Школу Космических Десантников, но по уровню боевой подготовки не уступавшая многим сослуживцам - мужчинам.

- Васкез, успокойся, - сказал Кренон, внимательно разглядывая противоположный берег. - Аборигенов надо отвлечь. Леон Троцкий возглавит группу, которая попытается обойти их с тыла. С ним пойдут Дон, Курт, Люк Соло и Эйра Кама. Пройдете направо вверх по ручью, перейдете на тот берег и атакуете туземцев с тыла. Джим, ты будешь прикрывать группу Леона с этого берега. Выбери удобную позицию. Я, Зира Кама и Хенли пойдём следом за ними и, в случае чего, поддержим огнем. Васкез, Сью и Джекки Доннелли - оставайтесь пока в укрытии и смотрите в оба, чтобы нас самих не обошли. Задача ясна?

Вопросов не было. Кренон хлопнул по плечу притаившегося рядом Кейлони:

- С Богом!

Дон короткими перебежками от куста к кусту стал пробираться к Леону Троцкому. За спиной, не сказав ни слова, прошмыгнул и Курт Левин.

Кренон подождал пока к нему подойдут Зира Кама и Хенли, а затем быстро перебежал ближе к ручью, который мирно журчал по ту сторону поляны. За ним последовали остальные. Выбрав удачную позицию за полуразвалившимся валуном, утопавшим в

болотной жиже, Кренон осмотрел противоположный берег ручья, основательно заросший высокой травой и кустарником. Пальмовая роща, из которой вылетела первая стрела, находилась левее, ниже по ручью. Ничего подозрительного там не наблюдалось. Видимо, туземцы были хорошими воинами и удачно использовали естественную среду для маскировки.

- Леон, отзовись, - Кренон не отрывал глаз от пальмовой рощи.

- Все в порядке, - доложил Троцкий, - Только что видели свампера. Быстро плавает, ничего не скажешь. Правда, мы его все равно обогнали. Так что, когда пойдете через ручей, будьте осторожны - дедушка крокодил остался голодным.

- Учтем обязательно, - ответил Кренон.

Тишина уже действовала на нервы. Казалось, что не было никакой атаки туземцев, не свистели ядовитые стрелы. Но внутреннее чутье подсказывало, что враг рядом и пристально следит за каждым их движением. Кренон посмотрел налево и увидел, как из плотной стены болотного тростника выскочил Джекки Доннелли и, чуть пригнувшись, побежал к берегу ручья. В этот момент капитан десантников и заметил аборигена. Сначала как-то подозрительно зашевелился развесистый куст на противоположном берегу прямо напротив пересекавшего поляну Джекки, а затем, словно из-под земли, появилось ОНО. Кренон находился в удобной позиции, под углом к существу, и ему удалось хорошо рассмотреть его. Высокого роста, передвигавшийся на двух мускулистых ногах, покрытых серой чешуей, монстр скорее походил на динозавра из-за своей продолговатой, с хорошо развитой челюстью, головы. В любой другой ситуации Кренон никогда бы не поверил, что перед ним разумное существо, но массивный лук в мускулистых руках и кожаный колчан, туго набитый отравленными стрелами, говорили сами за себя.

Абориген собирался стрелять по бегущему десантнику и Кренону требовалось во что бы то ни стало опередить его. Он прицелился в голову и в тот момент, когда монстр уже натягивал тетиву, выстрелил. Неприятель дернулся, словно от удара током, однако на ногах устоял и попытался спрятаться за кустом, но второй луч насквозь прожег покрытый чешуей череп. Кренон был неприятно удивлен увидев, что монстр не свалился после первого выстрела, а даже попытался скрыться. Такой степенью защиты не обладали даже бронезилеты землян. Кренон помахал рукой в ответ на благодарственное приветствие Доннелли, который успешно добрался до зарослей тростника у самого берега.

- Командир, - вышел на связь Троцкий, - я видел, как ты его уложил, но похоже это только начало.

Над болотами пронеслось эхо выстрелов. Кренон на слух определил, что стреляли из фотонного излучателя и из лазерного пистолета. Видимо, группа Троцкого столкнулась с аборигенами.

Под прикрытием Джекки Доннелли Кренон, Хенли и Зира Кана в мгновение ока перебрались на крутой берег.

Впереди за деревьями они увидели двух аборигенов, отступавших под мощным натиском десантников. Кренон и его партнеры находились в очень выгодной позиции и им не составило труда отправить незадачливых лучников к праотцам.

Когда все стихло капитан вышел в эфир:

- Леон, не стреляйте. Мы идем к вам.

Из-за деревьев появились фигуры десантников. Кренон направился к ним. На Леона Троцкого он сразу обратил внимание. Тот был явно чем-то озабочен и, когда капитан подошел к нему, Троцкий молча показал в сторону голубой полоски небольшой реки, которая отделяла их от... Впрочем, Кренон не сразу понял, что он там увидел. В редких просветах пальмового леса, начинавшегося сразу за рекой, были видны какие-то постройки и сооружения.

- Похоже, это их логово, - сказал Троцкий - Что будем делать?

- Вообще-то это уже не по нашей линии, - заметил Зира Кама. - Нет никакого смысла туда соваться.

- Надо идти. - Кренон был непоколебим. - И мы должны выяснить все, чтобы те, кто придет после нас, могли здесь спокойно работать.

Он включив передатчик:

- Васкез, как у вас?

- Тут пришли Диана, Хикс и Хадсон. Они услышали стрельбу и поэтому поспешили к нам.

- Командир, - вмешался капрал Хикс, - Мы можем чем-нибудь помочь?

- Вызывайте из лагеря остальных. Мы нашли логово этих папуасов. Все идем туда. Ваша группа пойдет по левому флангу через устье реки. Направление точно на север. Заодно прикроете нас. Старший группы капрал Хикс. Все, кто еще не перешел ручей, присоединяются к вам. Понятно?

Капрал Хикс не ответил. Вместо этого по лесу прокатилось эхо выстрелов.

- Я думаю, они разберутся, - пробормотал Кренон. - Пошли.

Где-то совсем рядом просвистела стрела. Он услышал сдавленный крик и обернулся. На траве без движений лежал окровавленный десантник. Это был Люк Соло. Стрела угодила ему в висок.

- Убит, - выдохнул Зара Кама, склонившись над телом.

Едва он успел произнести это слово, как над его головой, сшибая ветки кустов, пролетели пули. Зира плюхнулся на землю и замер. Вся группа затаилась в ожидании. На время все стихло. Кренон прятался за толстым стволом старой пальмы. Это было что-то новенькое: туземцы, как оказалось имели огнестрельное оружие. "Вот тебе и каменные топоры," - подумал командир десантников, прислушиваясь к каждому шороху.

- Одного я вижу, - прошептал стоящий рядом Дон Кейлони. - Попробую...

Он не договорил. На их группу снова обрушился град пуль и стрел. Аборигены засели у самого берега реки и, видимо, никого не собирались пропускать на другую сторону. Но Дону удалось использовать секундную паузу: его лазерное ружье выстрелило в тот момент, когда один из аборигенов попытался проскочить между деревьями. Выстрел оказался точным. Массивное тело монстра покачнулось и, под хруст ломаемых веток, завалилось на землю. Вдохновленные удачей Кейлони, десантники открыли огонь по зарослям тростника, за которым прятались туземцы. Под хорошим прикрытием лазерных лучей Курт Левин добрался до трупа аборигена, убитого Доном и там подобрал смертоносное оружие внеземного происхождения.

Когда десантники прекратили обстрел вражеских позиций, Курт показал оружие инопланетян Кренону.

- Так это же терминатор! - воскликнул капитан, взяв в руки ружье.

Курт утвердительно кивнул. Всем опытным десантникам было известно это новое оружие, которое по своим характеристикам могло сравниться, пожалуй, даже с фотонным излучателем. Терминатор стрелял разрывными пулями, наполненными кислотой. В магазин помещалось до двухсот пятидесяти патронов, что являлось неоспоримым преимуществом перед многими другими видами оружия. Помимо этого ружье имело оптический прицел с встроенным прибором ночного видения и еще массу всякого рода достоинств. Впервые терминаторы обнаружили на одной давно вымершей планете в системе Сириуса. Там был найден целый арсенал всякого рода вооружений. Цивилизация, оставившая это наследство, перестала существовать задолго до появления первого человека на Земле и, вообще-то, не достигла больших высот в освоении Вселенной из-за глобальной термоядерной войны, практически уничтожившей планету. Кто дал оружию название терминатор никто уже не помнил, но оно было во всех учебниках по вооружениям.

- Боюсь, что мы их не за тех приняли, - вслух подумал Кренон, отдавая инопланетное оружие Курту Левину.

- Насколько мне известно, в системе Сириуса цивилизация была гуманоидной, а эти монстры ничего общего с ними не имеют, - Курт повесил терминатор на плечо.

- А это уж пусть ученые решают, как терминаторы попали к этим... Кренон не нашел подходящего названия для существ, с которыми они столкнулись.

В продолжение этого разговора они двигались к реке, за которой находилось логово монстров. Несколько раз им пришлось вступать в перестрелку с туземцами, вооруженными терминаторами и луками. С лучниками все было достаточно просто, а вот с обладателями

оружия из системы Сириуса приходилось повозиться.

Когда они добрались до огромного каменного строения, изумлению их не было предела. Неправильной формы, с многочисленными окнами, походившими скорее на бойницы старинных крепостей, строение действительно напоминало приземистую крепость, каких на Земле, впрочем, никогда не строили. То, что издалека им казалось многочисленными постройками, на самом деле оказалось одним единственным домом, имевшим несколько прилегавших друг к другу флигелей. Ухоженная площадь перед логовом монстров была пустынной. Интересным было еще и то, что у этого дома не было дверей, в привычном понимании. Вместо них в стенах чернели проходы, похожие скорее на норы диких животных.

Группа капитана Кренона приблизилась к строению с правой стороны и, под прикрытием каменных стен, направилась ко входу. Остановившись возле норы, ведущей внутрь, Кренон вышел на связь с группой капрала Хикса.

- Нас окружили, - без предисловий сообщил Хикс. - Мне удалось прорваться и я сейчас иду к вам.

- Диана, - позвал Кренон. - Отзовись!

- Погибла Диана, - прохрипел Сью Шоу. Свамперы убили. Она была ранена и не успела перейти ручей... Васкез, прикрой меня.

Прогремели залпы далеких выстрелов. Всем было ясно, что группе капрала Хикса пришлось туго. Но лишь благодаря им Кренон и его партнеры получили возможность добраться до логова монстров.

Капитан дождал пока затихнут выстрелы, а затем снова вышел в эфир:

- Все кто меня слышат - отзовитесь.

- Дэниел Дар на связи. Мы с Шарки вызвали спасательный корабль. Он уже вылетел, сейчас идем к вам по вашему следу.

- Хорошо.

- Капитан, это Джим. Мы с Васкез и с Хадсоном пытаемся вырваться из окружения, как слышно? Прием.

- Слышу вас, Сью. Ты где?.. Прием.

Сью Шоу на связь не вышел. Напрасно Кренон вызывал его. Видимо ему так и не пришлось перейти злосчастный ручей. Никто больше не видел Сью с того рокового момента, когда он, под прикрытием Васкез, попытался прорваться сквозь окружение.

- Говорит Джекки Доннелли. У меня все в порядке, я подхожу к реке.

- Мы идем в гости к монстрам, - сообщил для всех Кренон и, повернувшись к десантникам, которые стояли рядом, добавил: - Ну что ж, посмотрим, как они нас встретят.

В небе сгущались сумерки, когда они вошли в дом. Прямо у входа их встречали два монстра с трезубцами в сильных руках. Впрочем, для десантников не составило особого труда быстро с ними расправиться. За спинами монстров прятались детеныши этих загадочных существ. Когда Кренон вошел в просторное помещение с фотонным излучателем наперевес, детеныши бросились врассыпную.

- Курт и Хенли налево, остальные за мной, - скомандовал Кренон.

Каменный пол и стены небольшого зала, в котором они оказались, были расписаны причудливыми узорами. Справа от них, у стены прямо сквозь пол рос небольшой куст какого-то растения с широкими плотными листьями, а на противоположной стороне они увидели два огромных яйца, из которых по-видимому и вылуплялись аборигены. Каждое яйцо лежало отдельно в небольшом углублении с подстилкой из сухой травы.

Осмотр зала занял всего несколько секунд, но как только десантники приблизились к своеобразному инкубатору, произошло то, чего так опасался Кренон. В широком проходе в соседнее помещение появилась громадная тварь, только отдаленно напоминавшая монстров, с которыми они воевали. Скрюченное туловище существа, покрытое серым панцирем, венчала гигантская голова с чешуйчатой мордой. Из продолговатого рыбьего рта торчали клыки, а большие выпуклые глаза малинового цвета зло смотрели из под свисавших мешками век. Уродливые тонкие руки были сложены на покатоной груди. Тварь передвигалась на двух жирных коротких ногах, похожих на широкие тумбочки. Вне всяких сомнений это

была matka монстров. Десантники открыли огонь. Лазерные лучи шипя вгрызались в прочный панцирь, оставляя на сером теле черные выжженные пятна. Но, похоже, мамаше этих тварей, лучи лазерных ружей и пистолетов не доставляли особых неудобств. Под лавиной огня matka ухитрилась несколько раз выплюнуть кислотной слюной, едва не зацепив кого-то из десантников. Каменная стена, в которую попала тварь, в считанные секунды превратилась в пыль. Их спасало только то, что расстояние, на которое долетала кислота, было небольшим и, выдерживая безопасную дистанцию, они достаточно спокойно вели обстрел. Положение осложнялось, однако, тем, что детеныши монстров не остались в стороне во время этой схватки, а бросились на помощь своей мамаше. Они были без оружия, но их крепкие зубы без особого труда могли перегрызть горло любому человеку. От клыков одного детеныша пострадал Хенли. Он перезаряжал пистолет, когда к нему сзади подкрался детеныш и, запрыгнув на спину десантнику, вцепился клыками в шею. От неминуемой смерти Хенли спас Леон Троцкий, одним выстрелом уложивший монстра. Битва закончилась неожиданно. После очередного залпа matka, обстрелянная со всех сторон, издала предсмертный крик и опрокинулась навзничь, звонко ударившись панцирем о каменный пол. Разобраться с отдельными защитниками королевы племени было уже нетрудно. Группа Кренона прошла вглубь опустевшего логова и с радостью встретила там до зубов вооруженных инопланетным оружием Хикса, Васкез, Хадсона и Джима Дигриза.

- А где Джекки Доннелли? - поинтересовался Кренон у Васкез.

- На корабле.

- За нами уже прилетели? - он выглянул в окно и действительно увидел стальную громадину корабля, удачно посаженного прямо у выхода из жилища монстров.

Кренон связался с Дэниелом, который вместе с Шарки в эти минуты уже пересек поляну перед домом и сейчас направлялся к кораблю. Дэниел ответил после недолгого молчания:

- Нам надо поторапливаться. Мы только что видели целый отряд туземцев, который пришел с юга. Они видимо вернулись с охоты и сейчас направляются сюда.

Как в подтверждение этих слов, со стороны реки прогремели выстрелы.

- Хватит нам бессмысленных смертей, всем быстро на корабль! - выкрикнул Кренон, подталкивая десантников к выходу.

Они пересекли просторный коридор и оказались в помещении расстрелянной королевы аборигенов. Здесь они снова увидели полуметровые яйца, в которых развивались зародыши монстров.

- Наверное их нужно прихватить с собой, - высказал мнение Дон Кейлони.

- Конечно, пусть ученые поломают головы, - поддержал его Троцкий.

Кренон с ними согласился и приказал взять с собой несколько яиц. Повесив ружья за спину, Дон, Хадсон и Зира Кама подняли по одному яйцу и последовали за всеми к выходу. А у корабля уже шла перестрелка и всем стало ясно, что выйти без боя из дома им не удастся.

У трапа корабля лежал сраженный пулями Лазарь Шарки. Раненый Дэниел Дар с перекошенным от боли лицом прятался за углом дома.

- Мы прикроем вас, - сказал Кренон застывшим в нерешительности десанникам, которые несли тяжелые яйца, подобранные в комнате матки.

Это были самые трудные минуты в его жизни. Один за другим десантники перебежали опасную зону, разделявшую выход и трап корабля.

На его глазах, не добежав до трапа двух шагов, замертво рухнул на землю Зира Кама, настигнутый пулей. Яйцо, выпавшее из его рук, осталось лежать на траве, мешая проходу другим. В эти последние минуты был ранен и сам Кренон. Выбросив бесполезный фотонный излучатель с опустевшим магазином, Элмер Кренон рванул к трапу. Пули зацепили его в тот момент, когда капитан бежал вверх по ступенькам. Все уже были на борту, и как только Кренон, тяжело дыша ввалился в люк, корабль стал медленно отрываться от земли.

Долго еще он не мог забыть эту высадку на Зеро. Иногда по ночам ему снились кошмарные сны, в которых он опять вел свой отряд в логово монстров, а из зарослей болотного тростника погибшие Зира Кама, Сью, Люк Соло и Лазарь Шарки укоризненно смотрели ему в спину и молчали. После этой истории Кренон ушел из отряда и подался на

небольшой частный корабль, капитан которого был дальним родственником Кренона. И долгие годы он провел, занимаясь межпланетной торговлей. Многих из своего отряда он больше так и не увидел. Кто-то погиб на далеких планетах, а некоторые, как и он, оставили службу и поселились в разных концах Галактики.

Впервые за последние годы Кренон выбрался, чтобы навестить своего друга Дона Кейлони, который к этому времени уже возглавлял Центр Космической Безопасности Солнечной Системы. Впервые он так отчетливо вспомнил события на Зеро и рассказал о них внуку. Но он не смог удовлетворить любопытства юноши, поскольку не знал всех подробностей расследования гибели профессора Джонсона.

- Дед, ну откуда тогда у этих монстров взялось инопланетное оружие?

- Одно время ходили слухи, что эти существа все-таки были дальними родственниками гуманоидов с Сириуса, но по каким-то причинам они пошли по другому пути развития. Возможно, главную роль здесь сыграла термоядерная война, разразившаяся на их родной планете.

- А как они тогда оказались на Зеро?

Кренон встал из кресла и, застегивая костюм, устало посмотрел на внука:

- Ты знаешь, спроси это лучше у дяди Дона, когда мы прилетим.

Он посмотрел на экран. Их корабль уже медленно входил в шлюзовые ворота стыковочного отсека станции "Венера - Центр"...

ПОСЛЕДНИЕ ИЗВЕСТИЯ

В эти октябрьские дни "ИНФОРКОМУ" исполнилось четыре года. В связи с этим мы предприняли ряд важных шагов, для которых, как нам кажется, настало время.

* * *

На днях принято решение (и соответствующие меры), чтобы "ZX-РЕВЮ" стал в будущем году полноценным изданием с полиграфической точки зрения. У него будет стандартный книжный формат, высококачественная печать, красочная обложка, большее количество страниц и значительно увеличится тираж. В отличие от прошлых лет мы не будем искусственно сдерживать темп его распространения. Стиль и содержание останутся теми же. В соответствии с пожеланиями читателей больше внимания будет уделяться вопросам музыки, графики и работы в дисковых системах. К концу года планируется начать освещение работы "Спектрума" в сетях.

Благодаря качественному полиграфическому исполнению каждого номера мы откажемся от традиционной практики выпуска в конце года годового сборника. Итак, годовой сборник "ZX-РЕВЮ-93" станет последним.

* * *

Объявленное в прошлом выпуске акционерное общество "ИНФОРКОМ-ПРЕСС" прошло государственную регистрацию и уже работает. Заканчивается подготовка первого номера электронного журнала "РС-РЕВЮ", посвященного чисто игровой тематике для РС-совместимых компьютеров. Объем, содержание и оформление этого журнала пока являются беспрецедентными для нашей страны.

Редакция нового журнала будет представлена читателям в первом выпуске, который пойдет в распространение 01.01.94г.

Журнал будет распространяться по каналам собственной сети региональных дистрибуторов фирмы без защиты от копирования. Подробности см. в "ZX- РЕВЮ" N 7-8 за этот год.

Подготовлены условия для продажи ограниченного количества экземпляров этого издания в подарочном исполнении (в красочном альбоме) через специализированные компьютерные салоны г. Москвы по престижно высоким ценам (порядка 7-12 \$ за номер). Эта мера принята для поддержки региональных дистрибуторов. Наличие в Москве престижно высокого ценового уровня будет способствовать успеху их деятельности в регионах.

* * *

Вышел из печати и поступил в рассылку владельцам именных сертификатов второй том "Графики Спектрума".

В ближайшие дни выходит из печати книга "30 часов БЕЙСИКА".

* * *

Заключено соглашение между фирмами "СЛОТ" и "ИНФОРКОМ" об эксклюзивном освещении в "ZX-РЕВЮ-94" дисковой операционной системы для "Спектрума" IskraDOS.

Разработан план совместных мероприятий по продвижению этой системы на рынки программных средств СНГ. Эти шаги согласованы с владельцем авторских прав на систему, компанией IskraSOFT, г. С.-Петербург и получили поддержку. (Фирма "СЛОТ", г.Москва, является Генеральным дистрибутором компании IskraSOFT).

Подготовлена и передана в печать книга, посвященная работе с этой системой.

Систему IS-DOS, пакеты прикладных программ, разработанные под нее и сопутствующую литературу можно будет приобретать как у фирм IskraSOFT и "СЛОТ", так и по каналам "ИНФОРКОМа".

Содержание

СПЕКТРУМ В ШКОЛЕ	1
SINCLAIR LOGO	8
7. И СНОВА ОБ ОБРАБОТКЕ СПИСКОВ.	8
Уровни процедур.	10
Глобальные и локальные переменные.	11
Команда OUTPUT для рекуррентных процедур.	12
Вложенные списки.	13
Процедуры создания списков	14
Сортировка чисел.	15
Самообучающаяся игра.	18
8. НЕКОТОРЫЕ МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ.	20
ИСТИНА и ЛОЖЬ (TRUE и FALSE)	21
Возведение в степень и извлечение корня.	22
Поиск простых чисел.	22
Тригонометрические операции.	23
ПРИМЕНЕНИЕ АССЕМБЛЕРА ДЛЯ СОЗДАНИЯ БЫСТРОРАБОТАЮЩИХ ПРОГРАММ	26
8. Команды ATTRIBUTE, SCREEN\$ и POINT.	26
8.1. ATTRIBUTE (строка, столбец).	26
8.2 SCREEN\$.	27
8.3 POINT.	29
9. ПРИНТЕР.	30
9.1. COPY.	30
9.2 LPRINT.	30
9.3 LLIST.	31
СТРАНИЦА IS-DOS	32
Первое знакомство с iS-DOS.	34
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	42
КОМПЬЮТЕР И ЗВУК	42
ЧИТАТЕЛЬ - ЧИТАТЕЛЮ	49
К ВОПРОСУ О СТАНДАРТИЗАЦИИ	49
СПЕССУ + АСТРОЛОГИЯ	57
Что такое гороскоп.	57
Основной расчет.	59
Календарь.	60
Ввод исходных данных.	61
Астрологические дома.	61
"Дебют".	62
Апология Бейсика.	65
Параметры орбит.	67
Положения Солнца и планет.	67
Луна.	69
Расчет астрологических домов по системе В. Коха.	71
Управляющая программа.	71
СДЕЛАЙТЕ САМИ	74
ADVENTURE GAMES	74
Введение.	74
Структура системы.	74
Листинги.	75
Работа с листингами.	84
Головная программа ABS.	85
Возможные перемещения.	86
Расположение объектов.	87
Загрузчик.	87
Демонстрационная программа.	87
КОМПЬЮТЕРНАЯ НОВЕЛЛА	88
СТРАТЕГИЯ КАПИТАНА КРЕНОНА	88