

6155

Экз. чит. зала

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

11 - 6155



ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

А.А.Карлов, Т.Ф.Смолякова

О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ
СВЕТОВОГО КАРАНДАША

1971

11 - 6155

А.А.Карлов, Т.Ф.Смолякова

О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ
СВЕТОВОГО КАРАНДАША

**Научно-техническая
библиотека
ОИЯИ**

Карлов А.А., Смолякова Т.Ф.

11-6155

О программном обеспечении светового карандаша

Рассматривается логика и возможности группы дисплейных подпрограмм, обеспечивающих работу светового карандаша в режимах указания и слежения для дисплея на ЭВМ СДС-1604А. Приводятся примеры программирования этих режимов в программах, написанных на языке ФОРТРАН.

Сообщения Объединенного института ядерных исследований

Дубна, 1971

Световой карандаш (СК) является одним из основных средств общения человека с машиной в системах графического взаимодействия "человек-дисплей-ЭВМ". Эффективность работы со световым карандашом определяется не только его конструктивными особенностями.

Она в основном зависит от того, насколько простым и в то же время гибким с точки зрения пользователя будет программирование процедур взаимодействия человека с изображением.

Как известно, возможны два режима работы светового карандаша: режим указания, когда человек с помощью СК указывает на уже существующие на экране объекты, и режим слежения, когда СК используется для построения новых объектов на экране дисплея, задания их геометрических параметров и топологии.

В режиме указания реакция системы на сигнал от СК определяется характером решаемой задачи и единственно, что требуется от дисплейных подпрограмм, обеспечивающих этот режим, - это сообщить программе пользователя о том, что такое указание имело место, а также что и где было указано. Понятно, что сам факт появления сигнала от СК, преобразованного, скажем, в сигнал прерывания ЭВМ, еще не означает, что об этом станет известно программе пользователя, написанной на некотором языке высокого уровня, например, на языке ФОРТРАН,

так как последнему недоступно состояние системы прерывания. Другими словами, нужен механизм передачи информации о поступлении сигнала от СК через переменные, выделенные пользователем для этой цели в своей программе.

В режиме слежения на экран дисплея должен выводиться специальный маркер той или иной формы. При программной реализации режима слежения соответствующие дисплейные подпрограммы обеспечивают перемещение маркера вслед за СК, фиксируя при необходимости координаты промежуточных положений маркера и высвечивая, если требуется, траекторию его движения. Это позволяет рисовать объекты на экране, непосредственно вводя информацию в ЭВМ световым карандашом.

Несмотря на то, что режим слежения фактически сводится к многократной обработке информации о том, насколько СК сместился относительно центра маркера, ряд требований, выдвигаемых при этом программой пользователя, приводит к тому, что логика работы подпрограмм слежения не является тривиальной.

В основном эти требования сводятся к следующему:

1. После вызова СК процесс слежения должен осуществляться одновременно и независимо от работы программы пользователя.
2. В любой момент программе пользователя должны быть доступны координаты текущего положения маркера на экране.
3. Необходимо обеспечить минимум усилий со стороны пользователя для задания типа слежения: без рисования или с рисованием. В последнем случае пользователь лишь указывает минимальный шаг, с которым должны фиксироваться координаты промежуточных положений маркера и вид аппроксимации (линейными отрезками, пунктирными линиями и т.п.). Сам процесс фиксации координат и высвечивание тра-

ектории движения СК, как процедура стандартная, должен быть возложен на служебные дисплейные подпрограммы.

4. Возможность присвоения отдельным частям рисуемого изображения персональных названий с тем, чтобы они могли в дальнейшем обрабатываться по тем же правилам, что и объекты, сформированные обычным путем.

5. Возможность устанавливать логическую связь между маркером и произвольным объектом на экране так, что последующее перемещение маркера вслед за СК вызвало бы соответствующее перемещение выбранного объекта. Такая возможность исключительно важна в задачах, связанных с построением на экране каких-то сложных объектов, состоящих из некоторых более простых типовых элементов, например, в задачах построения электрических цепей из набора электронных компонент.

Исходя из перечисленных выше требований, для разработанного в ОИЯИ и подключенного к ЭВМ CDC-1604A дисплея со световым карандашом была создана группа дисплейных подпрограмм: OUTLPEN, OUTCROSS, OUTDRAW, OUTSTRIP, LPINTR, OUTCUT, OUTCOPY.

Подпрограмма OUTLPEN предназначена для настройки недоступной пользователю подпрограммы обработки прерываний от светового карандаша (LPINTR) на работу с переменными пользователя. Обращаясь к этой подпрограмме, программист указывает переменные, которые он выделяет под флаг светового карандаша (LRF), название выбранного СК объекта (NAME), а также под координаты и код яркости помеченной СК точки (IX , IY , IB). Подпрограмма OUTLPEN устанавливает в нуль переменную LRF вне зависимости от ее предыдущего значения. Если в дальнейшем окажется, что

переменная LRF имеет ненулевое значение, это служит признаком того, что имело место прерывание от СК (в режиме указания) и что название выбранного СК объекта и параметры указанной СК точки присвоены соответствующим переменным пользователя.

Подпрограмма OUTCROSS служит для формирования маркера слежения в заданном месте экрана (IXC, IYC). Маркеру присваивается специальное название, по которому он отличается от других объектов на экране (OUTCRS). Адреса параметров IXC, IYC передаются в подпрограмму обработки прерываний от СК, что позволяет последней осуществлять непрерывное обновление в процессе слежения значений переменных IXC и IYC в соответствии с текущим положением маркера.

Тип слежения задается пользователем с помощью подпрограммы OUTDRAW. Первый параметр (LPMODE) определяет режим: без рисования (0), рисование точками (1), отрезками сплошных (2) или пунктирных (3) прямых. Второй параметр (ID) устанавливает, с каким минимальным шагом в единицах раstra необходимо фиксировать промежуточные положения маркера. Смещение относительно предыдущего зафиксированного положения на расстояние, меньшее заданного, игнорируется.

Подпрограмма OUTSTRIP позволяет пользователю указать название того объекта, который он хотел бы перемещать с помощью маркера (параметр NAM), а также координаты точки на экране (IXM, IYM), куда следует поместить маркер. Обычно, хотя и необязательно, здесь задаются координаты одной из точек перемещаемого объекта. Параметры подпрограммы OUTSTRIP также передаются подпрограмме обработки прерываний от СК, при этом параметр, определяющий тип

слежения, устанавливается равным 4, что позволяет отличить данный режим от остальных.

Рассмотрим работу подпрограммы обработки прерываний от СК, блок-схема которой показана на рис. I.

При поступлении прерывания от СК работа программы пользователя прерывается и запоминается состояние рабочих регистров ЭВМ. Обмен с дисплеем через буферный канал приостанавливается на время обработки прерывания. Это позволяет подпрограмме обработки прерывания путем опроса состояния канала и устройства управления дисплеем установить, выдача какой точки на экран вызвала появление сигнала от СК и выделить координаты и код яркости этой точки. Кроме того, этой подпрограмме доступна специальная рабочая ячейка, в которой подпрограмма регенерации изображения, организуя выдачу очередного объекта на экран через буферный канал ЭВМ, запоминает название, присвоенное этому объекту программистом.

Анализируя название, подпрограмма обработки прерывания от СК определяет, от какого объекта пришло прерывание: от маркера слежения или от обычного объекта. В последнем случае название выбранного СК объекта, координаты помеченной точки и код ее яркости присваиваются соответствующим переменным пользователя, флаг светового карандаша устанавливается равным I, восстанавливается содержимое рабочих регистров ЭВМ, и возобновляется регенерация изображения на экране за счет сброса сигнала прерывания в устройстве управления дисплеем. После этого управление возвращается основной программе.

Если прерывание пришло от маркера, то выполняется сравнение координат помеченной СК точки с текущими координатами центра маркера. При совпадении координат происходит переход на возобновление регенерации и восстановление рабочих регистров ЭВМ с возвратом на

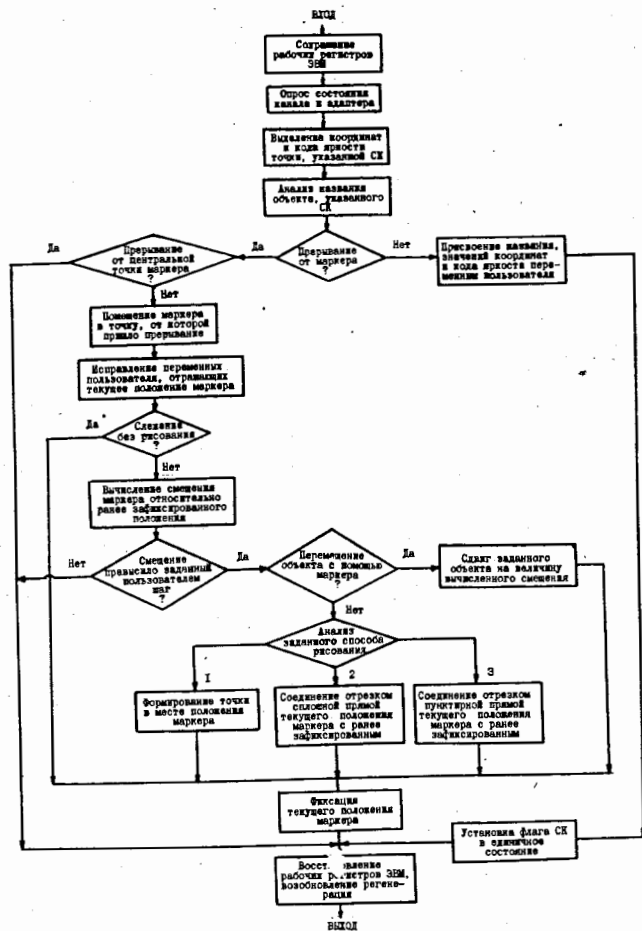


Рис. 1 Блок-схема подпрограммы LPRINTR

основную программу. Если же прерывание пришло не от центральной точки маркера, то с помощью дисплейной подпрограммы, обеспечивающей перемещение объекта с заданным названием в заданную точку на экране (OUTPLASE), выполняется перемещение маркера в точку, от которой пришло прерывание. Переменные, выделенные пользователем при обращении к подпрограмме OUTCROSS под координаты центра маркера, корректируются в соответствии с новым положением маркера, отражая его текущее положение. Затем производится анализ того, какой тип слежения был установлен пользователем. В случае слежения без рисования или если был задан режим рисования, но смещение маркера меньше минимального, установленного программистом, то после действий по возобновлению регенерации и восстановлению рабочих регистров ЭВМ следует возврат на основную программу. В противном случае в зависимости от выбранного типа слежения в текущем буфере формируется изображение точки или вектора (сплошного или пунктирного) в соответствии с новым положением маркера. Если через подпрограмму OUTSTRIP был выбран режим перемещения некоторого объекта по экрану с помощью маркера, то на данном этапе определяется величина смещения маркера относительно его предыдущего положения, и заданный объект смещается на ту же величину. Отметим, что прерывание от маркера не изменяет состояние флага светового карандаша.

Изображение, получаемое в процессе слежения с рисованием, становится видимым за счет того, что перед этим пользователь выделяет в качестве текущего буфера некоторую область, которая объявляется объектом с тем или иным названием. В результате подпрограмма регенерации начнет выдавать содержимое этой области на экран еще до начала рисования. Первоначальное содержимое этой

области - замаскированные точки, которые затем постепенно замещаются точками рисуемого объекта.

Упомянутое вначале требование о возможности присвоения частям рисуемого изображения отдельных названий достаточно просто удовлетворяется за счет специально созданной для этой цели административной дисплейной подпрограммы `OUTCUT`. В процессе рисования (рис.2) текущий буфер заполняется графической информацией. О необходимости объявить нарисованную часть изображения отдельным объектом сообщается программе пользователя, например, выбором соответствующей световой клавиши. В программе пользователя заполненная часть текущего буфера (длиной $L1$) объявляется объектом с названием `NAME1`, которое может быть введено, например, с клавиатуры или сформировано в самой программе пользователя. Затем объект, представляющий текущий буфер, с помощью подпрограммы `OUTCUT` укорачивается на величину $L1$.

Таким образом, пока рисуемому изображению не присвоено окончательное название, оно выдается на экран под названием, которое присвоил программист текущему буферу.

Другой полезной дисплейной подпрограммой при работе со световым карандашом в режиме слежения является подпрограмма `OUTCOPY`. Она относится к группе подпрограмм-генераторов и позволяет получить в текущем буфере копию заданного объекта. Например, в задаче построения некоторого сложного изображения из набора высвечиваемых на экране стандартных элементов требуется, чтобы при выборе СК одного из элементов оригинал оставался на месте, а вслед за маркером двигалась бы копия. Очевидно, что подпрограмма `OUTCOPY` существенно упрощает с точки зрения пользователя процедуру размножения объектов на экране.

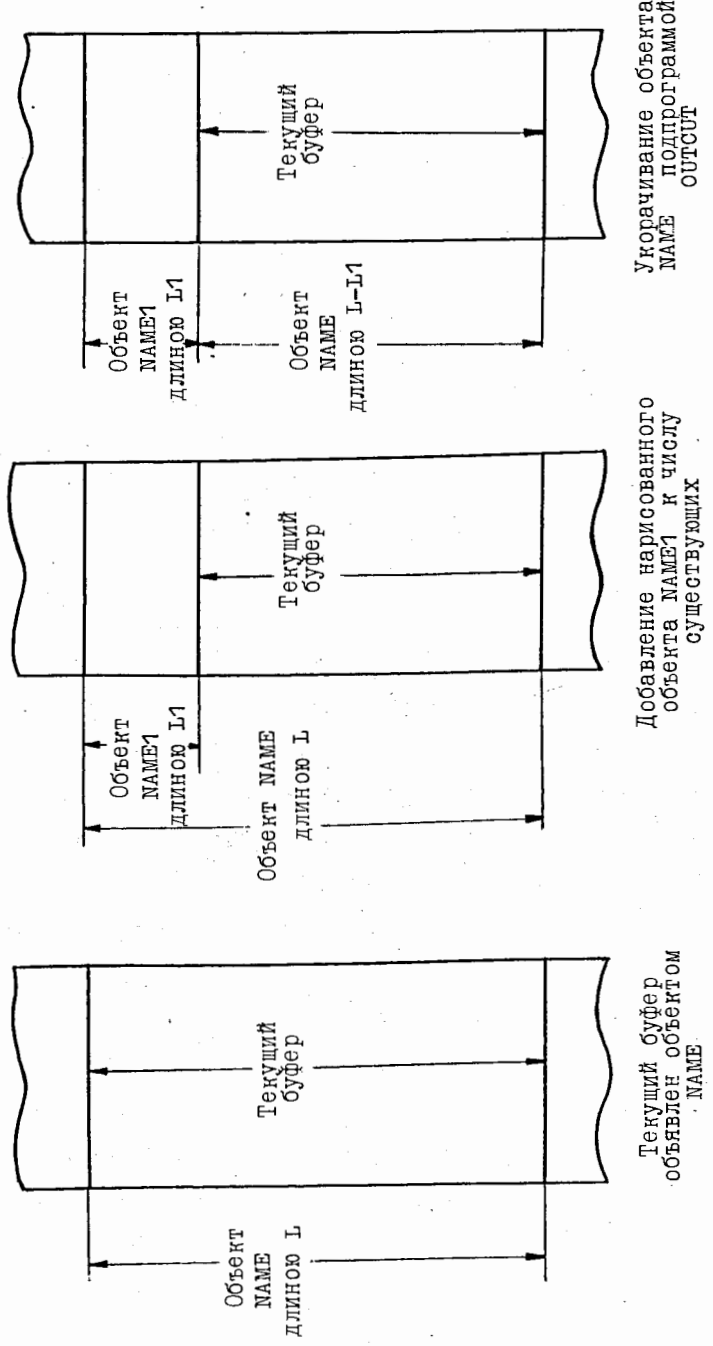


Рис.2. Формирование объектов в процессе слежения

На примере программ TRACE (приложение А) и BUILD (приложение Б) можно видеть, как пользоваться рассмотренными дисплейными подпрограммами при программировании на языке ФОРТРАН.

В программе TRACE после формирования в массиве IOSK набора световых клавиш и вызова маркера слежения (рис.3а) в буфере IBUF формируется и выдается на экран объект с названием TRASK, состоящий из 2000 замаскированных точек. Затем этот же буфер вновь объявляется текущим. В результате такой процедуры появление в дальнейшем в этом буфере какой-либо графической информации приведет к высвечиванию ее на экране. Затем следует обращение к подпрограмме OUTLREN и переход на цикл ожидания момента, когда флаг светового карандаша примет ненулевое значение (прерывания от маркера не изменят нулевого состояния флага).

Если сигнал СК поступил не от маркера, происходит выход из цикла ожидания, и начинается анализ названия выбранного СК объекта. Выбор клавиши PNT или CONT приводит к изменению типа слежения за счет обращения к подпрограмме OUTDRAW со значением первого параметра равным 1 (рисование точками) или 2 (рисование отрезками прямых) соответственно. Выбор клавиши OFF вызывает переход к режиму слежения без рисования. При этом ранее нарисованное изображение объявляется объектом с названием, равным значению переменной NAME1 в данный момент, с помощью дисплейной подпрограммы OUTFIND устанавливается длина (L) этого объекта, и объект TRASK укорачивается на величину L подпрограммой OUTCUT. Клавиша MARK позволяет отметить точкой текущее положение маркера на экране, а клавиша LINE - соединить отрезком прямой линии текущее положение маркера с ранее отмеченным. Клавиши

PX и PY позволяют строить на экране отрезки прямых параллельно координатным осям, при этом положение маркера соответствующим образом корректируется. Каждый раз при выборе клавиш LINE, PX и PY часть изображения, нарисованная в промежутке между выбором световых клавиш, объявляется отдельным объектом (рис.3б). Выбор клавиши DEL с последующим указанием СК на какую-либо часть нарисованного изображения приводит к уничтожению выбранного СК объекта.

На примере программы TRACE видно, что предлагаемый набор дисплейных подпрограмм для работы со световым карандашом требует от программиста лишь описания алгоритма взаимодействия человека с изображением в естественных для него понятиях переменных и освобождает его от необходимости самому осуществлять процесс слежения и фиксации промежуточных положений маркера, обеспечивать связь между системой прерывания и своей рабочей программой.

В подпрограмме BUILD показано использование подпрограмм OUTCOPY и OUTSTRIP для копирования и перемещения стандартных элементов (рис.4а,б). После выбора световой клавиши STRIP следует переход на цикл, где программа ожидает выборку световым карандашом какого-либо элемента. Если такая выборка имела место, то происходит копирование выбранного элемента через подпрограмму OUTCOPY, копия объявляется объектом с названием, равным текущему значению переменной NAME2, устанавливается логическая связь между маркером и копией через подпрограмму OUTSTRIP, и при этом маркер помещается в точку, от которой пришло прерывание. Назначение других световых клавиш очевидно из текста программы. В частности, клавиша DEL позволяет уничтожить любую копию стандартного элемента и только копию. Последнее обеспечивается проверкой названия объекта, выбранного вслед за клавишей DEL.

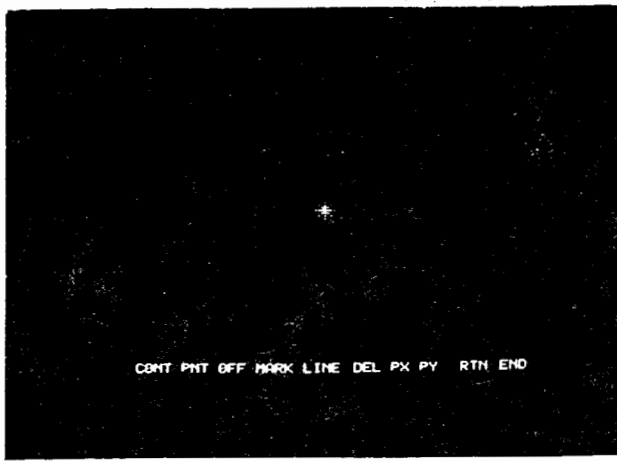


Рис. 3а Исходное изображение в программе TRACE

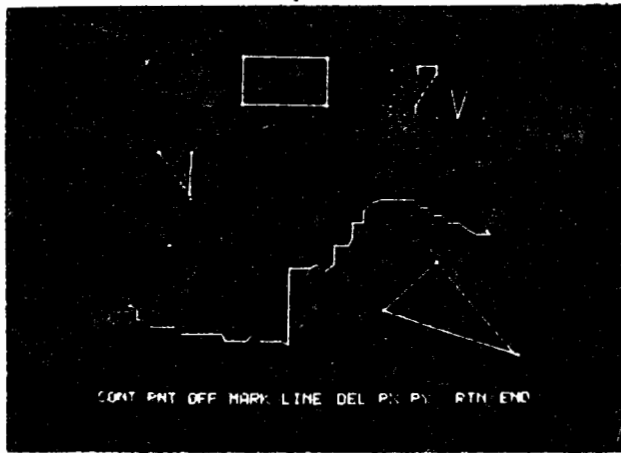


Рис. 3б Пример изображения, введенного посредством светового карандаша

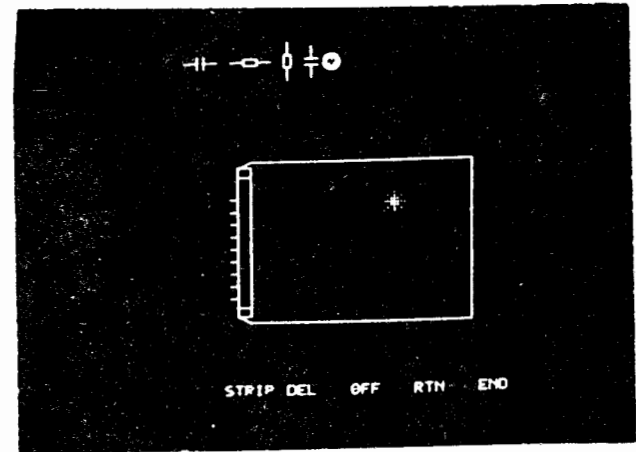


Рис. 4а Исходное изображение в программе BUILD

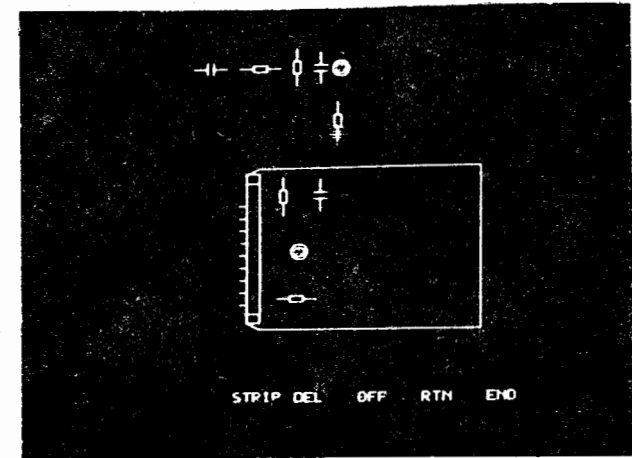


Рис. 4б Работа светового карандаша в режиме перемещения объектов по экрану дисплея

Пример программы BUILD еще раз подчеркивает необходимость идентификации объектов на экране с помощью названий. Действительно, положение копий стандартных элементов в произвольном случае нельзя предсказать заранее, а поэтому выполнение над ними каких-либо действий только на основании координат точки, помеченной СК, практически невозможно.

В заключение необходимо отметить, что реализация режима слежения программными средствами была обусловлена, с одной стороны, ограничениями на объем аппаратуры дисплея и сопряжения дисплея с ЭВМ, а с другой – достаточно высокой пропускной способностью буферного канала ЭВМ (50 кгц) и быстродействием машины (более 100 тыс. оп/сек). Последнее позволило получить скорость слежения до 5 см/сек, что вполне достаточно для большинства применений. Конечно, в случае сложных универсальных дисплеев с большим объемом электроники или в тех случаях, когда канал связи с ЭВМ не позволяет получить удовлетворительную скорость слежения, целесообразность аппаратурной реализации режима слежения не вызывает сомнений. Однако следует подчеркнуть, что взаимодействие человека с изображением отнюдь не сводится к перемещению маркера на экране какими бы то ни было средствами и в любом случае требует наличия дисплейных подпрограмм, обеспечивающих простую и вместе с тем эффективную связь программы пользователя с процессом слежения.

Рукопись поступила в издательский отдел
6 декабря 1971 года.

```

PROGRAM TRACE
COMMON/IOCK/ IOCK(1000), IBUF(1000)
98  CALL OUTCLEAR $ CALL OUTBUF(IOCK,500)
CALL OUTBCD(20,32,4HCONT,4,1) $ CALL OUTADDON(4HCONT)
CALL OUTBCD(140,32,3HPNT,3,1) $ CALL OUTADDON(3HPNT)
CALL OUTBCD(236,32,3HOFF,3,1) $ CALL OUTADDON(3HOFF)
CALL OUTBCD(332,32,4HMARK,4,1) $ CALL OUTADDON(4HMARK)
CALL OUTBCD(452,32,4HLINE,4,1) $ CALL OUTADDON(4HLINE)
CALL OUTBCD(578,32,3HDEL,3,1) $ CALL OUTADDON(3HDEL)
CALL OUTBCD(672,32,2HPX,2,1) $ CALL OUTADDON(2HPX)
CALL OUTBCD(744,32,2HPY,2,1) $ CALL OUTADDON(2HPY)
CALL OUTBCD(846,32,3HRTN,3,1) $ CALL OUTADDON(3HRTN)
CALL OUTBCD(948,32,3HEND,3,1) $ CALL OUTADDON(3HEND)
IXC=500 $ IYC=500 $ CALL OUTCROSS(IXC,IYC) $ ITR=1
ITF=1 $ NAME1=1 $ CALL OUTBUF(IBUF,1000)
DO 333 I=1,2000 $ CALL OUTDOT(0,0,0)
333 CONTINUE $ CALL OUTADDON(5HTRACK) $ CALL OUTBUF(IBUF,1000)
99  CALL OUTLPEN(LPF,NAME,IX,IY,IB)
100 IF(LPF.EQ.0) 100, 101
101 IF(NAME.EQ.4HCONT) 400, 301
301 IF(NAME.EQ.3HPNT) 399, 302
302 IF(NAME.EQ.3HOFF) 401, 303
303 IF(NAME.EQ.4HMARK) 402, 304
304 IF(NAME.EQ.4HLINE) 403, 305
305 IF(NAME.EQ.2HPX) 404, 306
306 IF(NAME.EQ.2HPY) 405, 307
307 IF(NAME.EQ.3HRTN) 98, 308
308 IF(NAME.EQ.3HEND) 777, 309
309 IF(NAME.EQ.3HDEL) 3090, 99
399 ITR=0 $ CALL OUTDRAW(1,15) $ GO TO 99
400 ITR=0 $ CALL OUTDRAW(2,15) $ CALL OUTDOT(IXC,IYC,1) $ GO TO 99
401 IF(ITR.EQ.1) 99, 4020
4020 IXMARK=IXC $ IYMARK=IYC $ CALL OUTDOT(IXC,IYC,1)
CALL OUTADDON(NAME1) $ CALL OUTFIND(NAME1,KA,L) $ NAME1=NAME1+1
CALL OUTCUT(5HTRACK,L) $ ITR=1 $ CALL OUTDRAW(0,0) $ GO TO 99
402 IXMARK=IXC $ IYMARK=IYC $ ITF=0
CALL OUTDOT(IXC,IYC,1) $ GO TO 99
403 IF(ITF.EQ.1) 99, 4030
4030 CALL OUTSHIFT(IXC,IYC,2) $ GO TO 200
404 IF(ITF.EQ.1) 99, 4040
4040 CALL OUTSHIFT(IXC,IYMARK,2) $ GO TO 200
CALL OUTPLACE(6HOUTCRS,IXC,IYMARK) $ GO TO 200
405 IF(ITF.EQ.1) 99, 4050
4050 CALL OUTSHIFT(IXMARK,IYC,2) $ IXC=IXMARK
CALL OUTPLACE(6HOUTCRS,IXMARK,IYC) $ GO TO 200
3090 CALL OUTLPEN(LPF,NAME,IX,IY,IB)
4000 IF(LPF.EQ.0) 4000, 4001
4001 IF(NAME.NE.4HCONT.AND.NAME.NE.3HPNT.AND.NAME.NE.3HOFF.AND.
1 NAME.NE.4HMARK.AND.NAME.NE.4HLINE.AND.NAME.NE.2HPX.AND.
2 NAME.NE.2HPY.AND.NAME.NE.3HRTN.AND.NAME.NE.3HEND.AND.
3 NAME.NE.3HDEL) 4010, 101
4010 CALL OUTDEL(NAME) $ GO TO 3090
200 CALL OUTADDON(NAME1) $ CALL OUTFIND(NAME1,KA,L)
CALL OUTCUT(5HTRACK,L) $ ITF=1 $ NAME1=NAME1+1 $ GO TO 99
777 CALL OUTCLEAR
END

```

PROGRAM BUILD

C

```

DIMENSION IOCK(2000)
DIMENSION NMS(5), ITS(6)
NMS(1)=5HSTRIP $ NMS(2)=3HDEL $ NMS(3)=3HOFF
  NMS(4)=3HRTN $ NMS(5)=3HEND
ITS(1)=4HRESV $ ITS(2)=3HRES $ ITS(3)=5HCONDV
  ITS(4)=4HCOND $ ITS(5)=5HPLATE $ ITS(6)=3HRTN

```

303

```

CONTINUE
CALL OUTCLEAR $ CALL OUTBUF(IOCK,2000)
  IX=130 $ IDX=150 $ IY=30
DO 70 I=1,5 $ CALL OUTBCD(IX,IY,NMS(I),5,1)
  CALL OUTADDON(NMS(I))

```

70

```

IX=IX+IDX
  CALL OUTCROSS(500,500)
CALL COND(32,890) $ CALL RES(144,890)
  CALL RES1(280,940) $ CALL CONDV(336,930)
  CALL TRN(384,890) $ CALL PLATE(192,224)
  NAME1=1

```

99

```

CALL OUTLPEN(LPF,NAME,IX,IY,IB)

```

100

```

IF(LPF) 101,100,101

```

101

```

IF(NAME.EQ.5HSTRIP) 200,102

```

102

```

IF(NAME.EQ.3HOFF) 304,104

```

104

```

IF(NAME.EQ.3HRTN) 303,103

```

103

```

IF(NAME.EQ.3HDEL) 305,105

```

105

```

IF(NAME.EQ.3HEND) 777,99

```

C

```

200 CALL OUTLPEN(LPF,NAME,IX,IY,IB)

```

300

```

IF(LPF) 301,300,301

```

301

```

IF(NAME.NE.5HSTRIP.AND.NAME.NE.3HOFF.AND.NAME.NE.

```

```

1 3HRTN.AND.NAME.NE.3HEND) 3010,101

```

3010

```

CALL OUTCOPY(NAME) $ CALL OUTADDON(NAME2)

```

```

CALL OUTSTRIP(NAME2,IX,IY) $ NAME2=NAME2+1 $ GO TO 99

```

304

```

CALL OUTDRAW(0,0) $ GO TO 99

```

305

```

CALL OUTLPEN(LPF,NAME,IX,IY,IB)

```

1000

```

IF(LPF.EQ.0) 1000, 1010

```

1010

```

DO 609 I=1,5 $ IF(NAME.EQ.NMS(I)) 99,609

```

609

```

CONTINUE

```

```

DO 619 I=1,6 $ IF(NAME.EQ.ITS(I)) 305,619

```

619

```

CONTINUE

```

620

```

CALL OUTDEL(NAME) $ GO TO 305

```

777

```

CALL OUTCLEAR

```

```

END

```