

АВТОКОД ДЛЯ РАБОТЫ С ГРАФИЧЕСКИМ ДИСПЛЕЕМ ЕС-7064

В.Л.Катков, Т.С.Янчук

Автокод предназначен для разработки программы, управляющих графическим дисплеем ЕС-7064 [1]. Хотя Автокод реализует все приказы дисплея, в том числе и те, которые организуют диалоговый режим работы, его применение ориентируется главным образом на пакетный режим, когда информация только выводится на дисплей: прерывания от светового пера, алфавитно-цифровой и функциональной клавиатур в первой реализации Автокода игнорируются. Одна из причин такого решения - отсутствие удовлетворительной операционной системы на ЭВМ ЕСМ-6, поддерживающей диалог с дисплеем.

При разработке Автокода одним из существенных моментов была стыковка вычислительной и графической частей программы: автокодские части должны органически входить в вычислительную программу, составляя с ней одно целое. В связи с этим целесообразно было ориентироваться на язык ФОРТРАН, который интенсивно используется для расчетов и работы с графическими устройствами.

Ниже приводится формальное описание языка, пример программы на Автокоде и ФОРТРАНе, реализация транслятора с Автокода.

§1. Описание языка

Всюду ниже при описании синтаксиса будут использоваться модифицированные бакусов-науровь формы; конструкция $\{x\}_1^n$ означает, что x не повторяется ни разу или один раз, $\{x\}_0^n$ - ни разу или любое число раз, $\{x\}_1^\infty$ - один или любое число раз. В некоторых случаях формулы могут заканчиваться ссылками на ФОРТРАН или АЛГАН без их дальнейшей конкретизации.

1.1. Основные символы.

Синтаксис:

$\langle \text{основной символ} \rangle ::= \langle \text{буква} \rangle | \langle \text{разделитель} \rangle$
 $\langle \text{буква} \rangle ::= A|B|V|Г|Д|E|Ж|З|И|Й|К|Л|М|Н|О|П|Р|С|Т|У|Ф|Х|Ц|Ч|Ш|Щ|$
 $|Ъ|Ы|Ь|Э|Ю|Я|G|I|J|L|N|Q|R|S|U|V|W|Z$
 $\langle \text{цифра} \rangle ::= 1|2|3|4|5|6|7|8|9|0$
 $\langle \text{разделитель} \rangle ::= \langle \text{знак} \rangle | \langle \text{скобка} \rangle | \langle \text{описатель} \rangle | \langle \text{конструктивный}$
 $\text{разделитель} \rangle$
 $\langle \text{знак} \rangle ::= +|-|/|'$
 $\langle \text{скобка} \rangle ::= (|)|\{|\}'$
 $\langle \text{конструктивный разделитель} \rangle ::= :|,|;|.|=$
 $\langle \text{описатель} \rangle ::= \text{ВОССТАНОВИТЬ|ЗАПОМНИТЬ.ЛУЧ|КАДР|КОНЕЦ.КАДРА |КО-}$
 $\text{НИЦ.ПРОЦЕДУРЫ|КОНЕЦ.РИСУНКА|ЛИНИЯ|ЛУЧ|ЛУЧ.МЕТКА|МАССИВ |НА|}$
 $\text{ПРОЦЕДУРА|РИСУНОК|СВЯЗЬ|СТЕРЕТЬ.ЛУЧ|СТРОКА|ТЕКСТ|ТОЧКА}$

Семантика:

Смысл и употребление основных символов поясняется в последующих разделах.

1.2. Идентификаторы, числа и строки.

Синтаксис:

$\langle \text{идентификатор} \rangle ::= \langle \text{буква} \rangle \{ \langle \text{буква} \rangle | \langle \text{цифра} \rangle \}^*$
 $\langle \text{число} \rangle ::= \{ + | - \}^* \langle \text{целое} \rangle$
 $\langle \text{целое} \rangle ::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}^*$
 $\langle \text{строка} \rangle ::= \langle \text{последовательность любых основных символов, отлич-}$
 $\text{ных от кавычек} \rangle$

Примеры

A
 АЛЬФА76
 38
 -001
 УГОЛ = 166, СИНУСАЛЬ = 5

Семантика:

Идентификаторы используются для образования переменных, имен массивов, имен процедур, меток и точек.

Числа могут быть только целыми.

Строки используются в основном для вывода символов с помощью оператора ТЕКСТ на экран дисплея.

1.3. Переменные.

Синтаксис:

$\langle \text{переменная} \rangle ::= \langle \text{скаляр} \rangle$
 $\langle \text{скаляр} \rangle ::= \langle \text{идентификатор} \rangle | \langle \text{переменная с индексом} \rangle$

< переменная с индексом > ::= < имя массива > (< индекс >)
 < имя массива > ::= < идентификатор >
 < индекс > ::= < индексное выражение ФОРТРАНА >

Примеры.

B
 B6
 A(52)
 A5(2 * K + I)

Семантика:

Переменные используются в операторах автокод-программы; переменные с индексом должны быть описаны соответствующим оператором ФОРТРАНА или Автокода.

1.4. Координаты.

Синтаксис:

< координаты > ::= < X-координата >, < Y-координата >
 < X-координата > ::= { 'X' }¹ < приращение >
 < Y-координата > ::= { 'Y' }¹ < приращение >
 < приращение > ::= < арифметическое выражение ФОРТРАНА >

Примеры.

+5, -6
 'X'5, 'Y' (B+I3) * A(52)

Семантика:

Координаты используются для задания позиции луча на экране дисплея: приращение указывает, на сколько растровых единиц^{*)} необходимо сместить луч от его текущего положения. Вторая конструкция ('X'..., 'Y'...) указывает абсолютные координаты новой позиции луча. Запрещено задавать одну координату как приращение, а вторую - абсолютно.

Значения координат луча не должны превосходить размеров экрана, т.е. должны изменяться от 0 до 1023.

1.5. Операторы.

Все действия в языке задаются с помощью операторов; оператор занимает, как правило, одну строку бланка.

1.5.1. Оператор МАССИВ.

Синтаксис:

< оператор МАССИВ > ::= МАССИВ < описание массива > { , < описание массива > }[∞]

^{*)} Растровая единица равна 0,25 мм.

< описание массива > ::= < имя массива > (< длина >) = < инициализация >
 < имя массива > ::= < идентификатор >
 < длина > ::= < целое >
 < инициализация > ::= < число > { , < число > }[∞]

Пример.

МАССИВ AI(3) = 5, -6, +1, A(3)=0

Семантика:

Оператор МАССИВ отводит в памяти линейную последовательность ячеек для элементов массива. Массивы могут быть только одномерными, элементы массива занумерованы числами натурального ряда, один элемент занимает одну ячейку памяти. Инициализация задает начальные значения элементов массива. Если число элементов инициализации меньше длины массива, то оставшиеся элементы массива ничем не заполняются (не определены); если число элементов инициализации больше длины массива, то массив инициализируется, а оставшиеся элементы инициализации игнорируются.

1.5.2. Оператор СТРОКА.

Синтаксис:

< оператор СТРОКА > ::= СТРОКА < описание строки > { , < описание строки > }[∞]
 < описание строки > ::= < имя строки > (< длина строки >) = < инициализация строкой >
 < имя строки > ::= < идентификатор >
 < длина строки > ::= < целое >
 < инициализация строкой > ::= < строка >

Пример.

СТРОКА П1(9) = ' _ _ _ _ И _ _ _ _ К А Д Р '

Семантика:

Оператор отводит в памяти линейную последовательность ячеек для строки. Заполнение ячеек происходит байтами слева направо. Если задана инициализация, то в отведенное место пишутся символы строки. В дальнейшем всю строку можно упоминать ее именем, например П1, а отдельные символы - переменной с индексами, например, П1(4) означает И. Оставшиеся незаполненными после инициализации позиции считаются пустыми.

1.5.3. Оператор ТОЧКА.

Синтаксис:

< оператор ТОЧКА > ::= ТОЧКА < описание точки > { , < описание точки > }[∞]
 < описание точки > ::= < имя точки > (< координаты >)

Пример.

ТОЧКА P (5,2), P4 ('X' B+2, 'Y' B-2)

Семантика:

Оператор связывает с именем точки ее координаты, заданные в приращениях или абсолютно. К моменту выполнения оператора должны быть вычислены значения всех переменных, входящих в изображение координат.

I.5.4. Оператор рисования точки.

Синтаксис:

< оператор рисования точки > ::= ЛУЧ.МЕТКА | ЛУЧ < список точек >
 < список точек > ::= < точка > { , < точка > }₀[∞]
 < точка > ::= { < режим > }₀¹ < обозначение точки >
 < режим > ::= < целое >
 < обозначение точки > ::= < имя точки > | < координаты > | (< X-массив > ,
 < Y-массив >)
 < X-массив > ::= { 'X' }₀¹ < имя массива >
 < Y-массив > ::= { 'Y' }₀¹ < имя массива >

Примеры.

ЛУЧ.МЕТКА

ЛУЧ P, (5,-2), P6

ЛУЧ O: (5,-2), ('X' B+2, 'Y' B-2)

ЛУЧ (A, A)

Семантика:

Выполнение оператора сводится к последовательному высвечиванию точек, перечисленных в списке. Режим задает яркость луча: 0 - яркость равна нулю (точка не светится, но луч устанавливается в указанную позицию); по умолчанию яркость считается стандартной, определяемой устройством ЕС-7064.

Последовательность точек можно задавать с помощью X-, Y-массивов, при этом длины массивов должны быть одинаковыми. Координаты рассматриваются либо как приращения, либо как абсолютные, если задана стандартная конструкция ('X'...., 'Y'....).

Оператор ЛУЧ.МЕТКА выполняет в дисплее приказ ВКЛЮЧИТЬ МЕТКУ, после чего луч выводится в центр экрана.

I.5.5. Оператор рисования линии.

Синтаксис:

< оператор рисования линии > ::= ЛИНИЯ < список точек >

Примеры.

ЛИНИЯ P, (5,-2), P6

ЛИНИЯ O: (5,-2), ('X' B+2, 'Y' B-2)

ЛИНИЯ (A, A)

Семантика:

Выполнение оператора аналогично рисованию точки, но только точки соединяются прямыми линиями при переходе луча от одной точки к другой. В первом примере будут соединены прямыми линиями точки P, (5,-2), P6. В случае X-, Y-массивов линия проводится, начиная с первой точки, заданной первыми элементами X-, Y-массивов, и кончая последней точкой.

I.5.6. Оператор перехода.

Синтаксис:

< оператор перехода > ::= НА < метка >
 < метка > ::= < целое без знака >

Пример.

НА 2

Семантика:

Оператор перехода передает управление по метке. Переход разрешается только внутри автокод-сегмента или внутри одного рисунка.

I.5.7. Оператор рисования текста.

Синтаксис:

< оператор рисования текста > ::= ТЕКСТ { < режим > }₀¹ < элемент текста > { , < элемент текста > }₀[∞]
 < элемент текста > ::= < строка > | < имя строки > | < шаблон > | < переменная >
 < шаблон > ::= # < как формат в АЛГАНСе >

Примеры.

ТЕКСТ 'УГОЛ=90'

ТЕКСТ I: 'УГОЛ' = ', 'z4d2b', r, 'ГРАДУСОВ'

ТЕКСТ I: II

Семантика:

Оператор рисования текста, начиная с текущей позиции луча, рисует текст, заданный строкой или именем строки. Режим задает способ выполнения оператора:

0 - символы имеют стандартный размер, ввод с алфавитно-цифровой клавиатуры (АЦК) в область текста запрещен;

I - символы имеют большой размер, ввод - запрещен;

2 - символы имеют стандартный размер, ввод с АЦК разрешен;

3 - символы имеют большой размер, ввод с АЦК разрешен.

По умолчанию предполагается 0-й режим.

Текст, подлежащий выводу на экран, можно редактировать с помощью символа '/', который означает перевод луча на следующую строку, причем начало строки определяется X-позицией луча, заданной до выполнения оператора ТЕКСТ.

Использование шаблона позволяет рисовать числовые значения в желаемом формате, как в АЛГАНСе.

1.5.8. Гнездо операторов.

Синтаксис:

< оператор начала гнезда > ::= ЗАПОМНИТЬ.ЛУЧ { < точка > }¹
< оператор конца гнезда > ::= СТЕРЕТЬ.ЛУЧ
< оператор возврата > ::= ВОССТАНОВИТЬ

Пример:

ЗАПОМНИТЬ.ЛУЧ (5,-2)
ВОССТАНОВИТЬ
СТЕРЕТЬ.ЛУЧ

Семантика:

Оператор начала гнезда запоминает координаты текущей позиции луча (или координаты заданной в нем точки) и выполняет операторы гнезда вплоть до закрывающей скобки СТЕРЕТЬ.ЛУЧ. Дожда до нее, оператор вернет луч в запомненную ранее позицию и перейдет к следующему оператору.

Оператор ВОССТАНОВИТЬ может встретиться только внутри гнезда, и его выполнение сводится к возврату луча в запомненную позицию.

Запрещается пересечение гнезд операторов, вложенность разрешается. Оператор перехода может передавать управление другому оператору только внутри гнезда (без входа во внутреннее гнездо или выхода наружу).

С помощью гнезда операторов удобно программировать сложные изображения, имеющие иерархическую структуру.

1.5.9. Процедуры.

Синтаксис:

< описание процедуры > ::= ПРОЦЕДУРА < имя процедуры > { (< формальный параметр > ; < формальный параметр >)⁰ }¹ < тело процедуры > < конец описания процедуры >
< имя процедуры > ::= < идентификатор >
< формальный параметр > ::= < идентификатор >
< тело процедуры > ::= < последовательность операторов >

< конец описания процедуры > ::= КОНЕЦ.ПРОЦ { < имя процедуры > }¹
< оператор процедуры > ::= < имя процедуры > { (< фактический параметр > ; < фактический параметр >)⁰ }¹
< фактический параметр > ::= < сбалансированный текст >

Примеры.

Описание процедуры
ПРОЦЕДУРА П (Т1; Т2)
ЛУЧ Т1
ТЕКСТ Т2
КОН.ПРОЦ П
Оператор процедуры:
П ((5,-2); УГОЛ = 90)

Семантика:

Описание процедуры связывает с именем процедуры (и ее формальными параметрами) тело процедуры и не задает никаких действий.

Оператор процедуры выполняется следующим образом: фактические параметры текстуально подставляются на место оператора процедуры, и выполнение тела начинается с его первого оператора.

Сбалансированность текста, подставляемого в качестве фактического параметра, означает сбалансированность в нем числа открывающих и закрывающих скобок. Кроме того, налагаются обычные требования равенства числа фактических и формальных параметров. Запрещено рекурсивное описание и использование процедур. Наконец, в теле процедуры не разрешается располагать описание других процедур.

Система может иметь библиотечные процедуры. В этом случае, если в программе встречается обращение к библиотечной процедуре, ее тело вставляется в программу при трансляции. Список библиотечных процедур определяется конкретной реализацией.

1.5.10. Операторы КАДР и КОНЕЦ.КАДРА

Синтаксис:

< оператор КАДР > ::= КАДР { < имя кадра > }¹ (< размер кадра > { < рамка > }¹ { , < координаты кадра > }¹)
< имя кадра > ::= < строка >
< размер кадра > ::= < ширина > , < высота >
< рамка > ::= < целое >
< координаты кадра > ::= < скаляр > , < скаляр >
< ширина > ::= < скаляр >

< высота > ::= < скаляр >
 < оператор КОНЕЦ.КАДРА > ::= КОНЕЦ.КАДРА < имя кадра >

Пример.

КАДР 'ФИЛЬМ _ 0 _ БЭСМ 6' (25,14)
 КОНЕЦ.КАДРА

Семантика:

Оператор КАДР должен предшествовать любой последовательности операторов, содержащей операторы Автокода. Выполнение очередного оператора КАДР приводит к стиранию старого дисплейного файла и формированию нового изображения.

Имя кадра выводится над левым верхним углом кадра. Если задана рамка, то формируемое изображение будет очерчено. Координаты кадра указывают положение на экране графического дисплея левого нижнего угла кадра. Оператор КОНЕЦ.КАДРА завершает последовательность операторов Автокода и приводит к выводу накопленного изображения в дисплей.

Описанная пара операторов является скобками, в которые необходимо заключать любой автокод-сегмент. Вложенность и пересечение кадров запрещаются.

1.5.11. Операторы РИСУНОК и КОНЕЦ.РИСУНКА

Синтаксис:

< оператор РИСУНОК > ::= РИСУНОК
 < оператор КОНЕЦ.РИСУНКА > ::= КОНЕЦ.РИСУНКА

Семантика:

Эта пара операторов является скобками, в которые необходимо заключать операторы, формирующие сменную часть изображения. Оператор РИСУНОК вызывает вывод уже накопленной части изображения в дисплей и фиксирует начало сменной части изображения. Оператор КОНЕЦ.РИСУНКА вызывает вывод сменной части изображения в дисплей и уничтожает его в памяти ЭВМ. Вывод следующего рисунка сотрет из изображения предыдущего на экране дисплея.

Приведем пример автокод-сегмента, поясняющий работу операторов КАДР, РИСУНОК, КОНЕЦ.КАДРА и КОНЕЦ.РИСУНКА.

КАДР (А, В, I, С, Д)] Начинается кадр (без имени) размерами А x В, координаты левого нижнего угла (С, Д).] Кадр очерчивается.
ЛУЧ 0: (X, Y)	
ТЕКСТ 0: 'ПОСТ. ЧАСТЬ' /] Луч выведен в позицию (А+Х, В+У).
ИЗОБР.	

ЛУЧ 0: ('X' X2, 'Y' Y2)] Луч выведен в позицию (X2,Y2)
РИСУНОК] Начинается сменная часть кадра
ТЕКСТ 0: РИС. _ I] Тело рисунка
КОНЕЦ.РИСУНКА] Конец сменной части. На экране появились рамка и два текста: 'ПОСТ. ЧАСТЬ ИЗОБР.' и 'РИС. _ I'
РИСУНОК] Начинается сменная часть кадра (затирает старую сменную часть)
ТЕКСТ 0: РИС. _ 2] На экране текст 'РИС. _ I' заменяется текстом 'РИС. _ 2'
КОНЕЦ.РИСУНКА	
ТЕКСТ 0: 'РАБОТА _ ОКОНЧЕНА'] На экране текст 'РИС. _ 2' заменяется текстом РАБОТА ОКОНЧЕНА
КОНЕЦ.КАДРА	

1.5.12. Оператор СВЯЗЬ.

Синтаксис:

< оператор СВЯЗЬ > ::= СВЯЗЬ < связанный массив ФОРТРАНА > (, < сменная часть массива ФОРТРАНА >)
 < связанный массив ФОРТРАНА > ::= < имя массива > (< длина массива >)
 < длина массива > ::= < арифметическое выражение >

Пример.

СВЯЗЬ A(I7), Z (K+I)

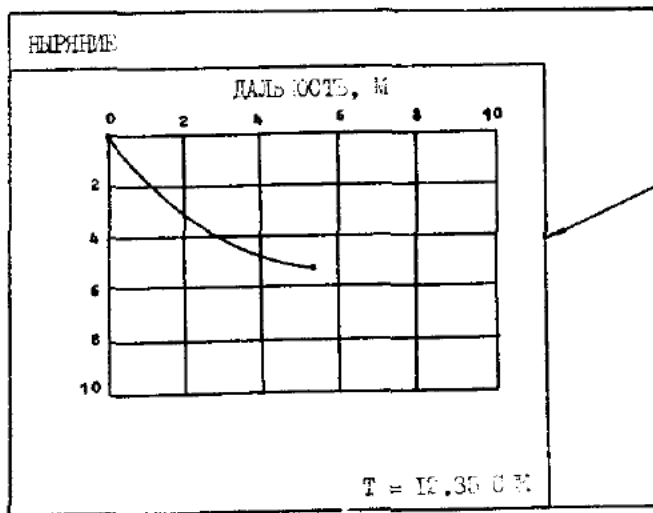
Семантика:

Оператор СВЯЗЬ должен предшествовать тем операторам Автокода, среди параметров которых есть массивы ФОРТРАНА. Он позволяет обходиться без явного задания длины массива в каждом операторе Автокода.

§2. Пример программы

Приводится пример программы вывода на экран дисплея изображения, получающегося в результате решения задачи о движении материальной точки под действием сил тяжести и сопротивления (задача о иррании). Изображаются траектория движения точки (переменная часть кадра) и координатная сетка с надписями (постоянная часть кадра). Схематическое изображение одного из кадров в некоторый момент времени приведено на рисунке.

Программа пишется на фортрановских бланках, операторы Автокод помечаются буквой А в колонке комментариев. Для сокращения записи существенная часть программы, связанная со счетом траек-



Кадр фильма о нырянии

тории, опущена; в массивах X, Y хранятся координаты точки на очередной момент времени, всего N пар координат.

§3. Реализация

3.1. Общая схема работы системы. На вход мониторной системы ДУБНА поступает задание, содержащее программы, написанные на ФОРТРАНЕ и Автокоде. Автокод-сегмент может быть вложен в любую из подпрограмм (SUBROUTINE) или в главную программу (PROGRAM).

Обработка задания происходит в два этапа. На первом этапе операторы Автокода транслируются в операторы ФОРТРАНА, адекватные им по исполнению (этап приведения к ФОРТРАНу). Здесь же происходит синтаксический анализ автокод-сегментов методом синтаксических подпрограмм, и на печать выдается соответствующая диагностика.

На втором этапе задача передается мониторной системе ДУБНА. Исполнение подпрограмм-образов операторов Автокода вызывает компиляцию дисплейного файла и вывод его на экран графического дисплея.

Дата _____
Лист _____

ФОРТРАН

Программа _____
Частота _____ сек.
С-Дискомментария _____

N	ОПЕРАТОР ФОРТРАНА	80
1	PROGRAM NYRNI	72
2	DIMENSION X(1000), (1000)	
3	ЗДЕСЬ ДОЛЖНЫ БЫТЬ ФОРТРАН - ОПЕРАТОРЫ	
4	ВЫЧИСЛЕНИЕ КООРДИНАТ ТРАЕКТОРИИ	
5	N = 20	
6	DT = 0.01	
7	ПЕРЕСЧЕТ КООРДИНАТ ИЗ ФИЗИЧЕСКИХ	
8	ЕДИНИЦ В РАСТРОННО ЕДИНИЦ ДИСПЛЕЯ	
9	DO 7 I = 1, N	
10	X(I) = X(I) * 94.	
11	Y(I) = Y(I) * 94.	
12	КАДР 'НЫРЯНИЕ' (1000, 1000, I)	
13	PRINT* ОСИ КООРДИНАТ С НАЛИЦЕМИ	

Программа _____
 Программист _____
 Институт _____ Тел. _____
 с. Д. Зв. Коммунарки

ФОРТРАН

Дата _____
 Лист _____

Линия	ОПЕРАТОР ФОРТРАНА	Кол-во
A	ЛУЧ 0 : ('X' 800 , 'Y' 965)	
A	ТЕКСТ 0 : ' ДАЛЬНОСТЬ , -М '	
A	ЛУЧ 0 : ('X' 40 , 'Y' 970)	
	DO 1 I = 1,6	
	R = (I - 1) * 2 .	
A	ТЕКСТ 0 : 'ZD' R	
A	ЛУЧ 0 : (I88 , 0)	
1	CONTINUE	
A	ЛУЧ 0 : ('X' 40 , 'Y' 965)	
	DO 2 I = 1,3	
A	ЛИНИЯ (940 , 0), 0 : (0, -I88), (-940, 0), 0 : (0, -I88)	
2	CONTINUE	
A	ЛУЧ 0 : ('X' 40 , 'Y' 965)	
	DO 3 I = 1,3	
A	ЛИНИЯ (0, -940), 0 : (I88, 0), (0, 940), 0 : (I88, 0)	
3	CONTINUE	
A	ЛУЧ 0 : ('X' 35 , 'Y' 965)	
	DO 4 I = 1,6	
	R = (I - 1) * 2	
A	ТЕКСТ 0 : 'ZD' R	

Программа _____
 Программист _____
 Институт _____ Тел. _____
 с. Д. Зв. Коммунарки

ФОРТРАН

Дата _____
 Лист _____

Линия	ОПЕРАТОР ФОРТРАНА	Кол-во
A	ЛУЧ 0 : (0 , -I88)	
C	CONTINUE	
C	РИСОВАНИЕ ТРАЕКТОРИИ	
C		
	DO 5 I = 1, N	
	ВРЕМЯ = (I - 1) * DT	
A	РИСУНОК	
A	ЛУЧ 0 : ('X' 40 , 'Y' 965)	
	DO 6 K = 1, I	
A	ЛИНИЯ ('X' 40 + X(K), 'Y' 965 - Y(K))	
C		
C	ВЫВОД ТЕКУЩЕГО ВРЕМЕНИ	
C		
A	ЛУЧ 0 : ('X' 500 , 'Y' 30)	
A	ТЕКСТ 0 : 'T = 'ZD , '2D' , ВРЕМЯ , -СЕК.'	
A	КОМАНДА РИС	
5	CONTINUE	
	STOP	
	END	

Приведение к ФОРТРАНУ осуществляется препроцессором, который просматривает исходную программу и для каждого оператора Автокода вызывает соответствующую ему подпрограмму. Эта подпрограмма анализирует синтаксис оператора, выделяет лексемы и по ним строит фортрановский образ оператора. Образ может содержать не только выполняемые, но и декларативные операторы. Выполняемые операторы образа вставляются на место своего прообраза, декларативные — добавляются к декларативным операторам объемлющей подпрограммы ФОРТРАНа введ за операторами PROGRAM или SUBROUTINE .

Каждая последовательность операторов, заключенная между операторами PROGRAM или SUBROUTINE и END , обрабатывается отдельно. Затем области декларативных и выполняемых операторов сочленяются, и начинается обработка следующей подпрограммы. Результат ее обработки добавляется к уже накопленному массиву. Процесс заканчивается формированием задачи, вызывающей мониторную систему ДУБНА.

Среди выполняемых операторов, которые являются образами операторов автокода, содержатся вызовы подпрограмм, реализующих работу с ЕС-7064. Таким образом, система состоит из препроцессора и библиотеки служебных подпрограмм этапа исполнения.

3.2. Представление объектов Автокода в ФОРТРАНе. Именованные объекты Автокода, задаваемые операторами МАССИВ, СТРОКА и ТОЧКА представляются в ФОРТРАНе целыми переменными с теми же именами, что и в Автокоде. Переменная "имя объекта" получает в качестве значения номер сопоставленного ей элемента таблицы атрибутов ТАТР. Кроме того, для массивов и строк заводится таблица значений ТЗ, в которой хранится их инициализация. Элемент ТАТР для точки имеет вид:

Признак точки	Признак абсолютности	X-координата	Y-координата
---------------	----------------------	--------------	--------------

Элемент ТАТР для массива (строки):

Признак массива	Длина массива	Ссылка в ТЗ на инициализацию
-----------------	---------------	------------------------------

Препроцессор программирует способ заполнения ТАТР и инициализацию ТЗ. Заполнение таблицы ТАТР происходит на этапе исполнения программы.

Кроме массивов ТАТР и ТЗ, необходимых для реализации операторов ТОЧКА, СТРОКА и МАССИВ, препроцессором генерируются следующие описания:

а) для операторов гнезда — массив ГНСТЕК, организованный по принципу стека, в который упрятываются координаты запоминаемой точки;

б) для метки и оператора перехода — массивы ТП и ТМ, по которым вычисляется адрес перехода;

в) для рисующих операторов ЛУЧ, ЛИНИЯ, ТЕКС и операторов управления процессом рисования КАДР, РИСУНОК, КОНЕЦ КАДРА и КОНЕЦ РИСУНКА — массив ДБ (дисплейный буфер), в котором накапливается информация для передачи в МОЗУ графического дисплея. Длина ДБ определяется длиной МОЗУ дисплея, длины прочих таблиц вычисляются препроцессором.

Таблицы, используемые во многих подпрограммах, описываются в COMMON-блоках. Для работы с таблицами вводятся целочисленные счетчики: INTEGER <счетчик 1>, <счетчик 2>, ..., <счетчик N>. Инициализация массивов и строк отображается в блок данных:

```

BLOCK DATA
COMMON /T3/T3(L)
DATA (T3 = < инициализация массива или строки 1 >,
      < инициализация массива или строки 2 >,
      . . . . .
      < инициализация массива или строки e > )
END
  
```

При этом, если число элементов инициализации массива (или строки) меньше длины, то оставшиеся элементы массива инициализируются нулями (оставшиеся элементы строки инициализируются пробелами).

Для слежения за положением луча на экране в общем блоке памяти описываются две целочисленные переменные, хранящие "текущие" значения его координат (ТКХ, ТКУ). Каждый рисующий оператор автокода пересчитывает ТКХ и ТКУ. Если при исполнении оператора луч выйдет за пределы кадра, то этот оператор исполняется частично, в пределах кадра, переменные же ТКХ и ТКУ получают значения предполагаемых координат луча.

Для контроля за положением луча в кадре описываются четыре переменные, хранящие границы изменения текущих координат, которые определяются параметрами оператора КАДР: Х1, У1 (координаты левого нижнего угла кадра), Х2, У2 (координаты правого верхнего угла кадра). Они также описываются в общем блоке памяти.

ЗАМечАНИЕ. Если программа не содержит операторов, наличие которых обуславливает генерацию описания массивов и переменных, то эти описания препроцессором не генерируются. Например, отсутствие операторов МАССИВ и СТРОКА позволяет не генерировать описание массива ТЗ и его счетчика.

3.3. Представление других операторов Автокода в ФОРТРАНе. Операторы Автокода транслируются в операторы присваивания и операторы вызова служебных подпрограмм. Вызов той или иной подпрограммы зависит не только от оператора Автокода, но и от вида его параметров. Соответствие операторов Автокода и служебных подпрограмм ФОРТРАНа приводится ниже в таблице.

ЗАМЕЧАНИЕ. Операторы Автокода, не рассмотренные в таблице, либо не имеют прямых фортрановских образов и отображаются в совокупности с другими операторами (например, оператор СВЯЗЬ), либо реализуются операторами присваивания.

Подпрограммы этапа исполнения выполняют следующие действия: МАССИВ, СТРОКА и ТОЧКА упаковывают атрибуты объекта в первый свободный элемент ТАТР.

ЛУЧК и ЛИНИЯК записывают в ДБ команды перевода луча из текущего положения в точку с заданными координатами в точечном или векторном режиме соответственно с заданной яркостью свечения.

ЛУЧИМ и ЛИНИМ работают аналогично ЛУЧК и ЛИНИЯК, выбирая координаты точки из ТАТР.

ЛУЧАМ и ЛИНИЯМ записывают в ДБ команды перемещения луча последовательно во все точки, определяемые значениями элементов Автокод-массивов.

ЛУЧФОМ и ЛИНФОМ отличаются от ЛУЧАМ и ЛИНИЯМ тем, что строят изображение по двум массивам ФОРТРАНа.

ЛУЧМЕТ выводит луч в середину экрана.

РИСТРО записывает в ДБ команды рисования строки, заданной именем, извлекая ее из ТЗ.

РИСТЕК отличается от РИСТРО тем, что строка является параметром подпрограммы.

Соответствие операторов Автокода и библиотечных подпрограмм этапа исполнения

Оператор Автокода	Вид параметров оператора Автокода	Библиотечная подпрограмма
МАССИВ	Массив Автокода	МАССИВ
СТРОКА	Строка Автокода	СТРОКА
ТОЧКА	Точка Автокода	ТОЧКА
ЛУЧ и ЛИНИЯ	Точка, заданная именем Точка, заданная координатами Два массива Автокода (массив точек)	ЛУЧИМ и ЛИНИМ ЛУЧК и ЛИНИЯК ЛУЧАМ и ЛИНИЯМ
ЛУЧ.МЕТКА	Два массива ФОРТРАНа	ЛУЧФОМ и ЛИНФОМ
ТЕКСТ	Без параметров Строка Автокода, заданная именем Строка Автокода Шаблон и переменная	ЛУЧМЕТ РИСТРО РИСТЕК РИСШАБ
ЗАПОМНИТЬ.ЛУЧ	Без параметров Точка, заданная именем Точка, заданная координатами	ЗАПЛУЧ ЗАПТОЧ ЗКООРД
ВОССТАНОВИТЬ.ЛУЧ	Без параметров	ВОССТ
СТЕРЕТЬ.ЛУЧ	Без параметров	СТЕЛУЧ
КАДР		РАМКА, ЛУЧК, РИСТЕК
РИСУНОК		РИС
КОНЕЦ.РИСУНКА		МЕТКА, ПОКАЗ, СТЕРИС
КОНЕЦ.КАДРА		МЕТКА, ПОКАЗ

РИСШАБ приводит переменную к строковому виду в соответствии с шаблоном и обращается к РИСТЕК.

ЗАПЛУЧ упрямляет текущие координаты луча в стек ГНСТЕК.

ЗАПТОЧ упрямляет координаты точки, заданной именем, в стек ГНСТЕК.

ЗКООРД упрямляет координаты в ГНСТЕК.

ВОССТ переводит луч в точку с координатами, которые последними были записаны в ГНСТЕК.

СТЕЛУЧ делает то же, что и ВОССТ, но стирает последнюю запись в ГИСТЕК.

РАМКА вычисляет значения ХИ, УН, МИ, УВ, чистит ДБ и, если нужно, записывает в ДБ команды очерчивающих рамки.

МЕТКА вычисляет адреса переходов и вставляет их в "дыры", приготовленные образом оператора НА.

ПОКАЗ передает элементы ДБ в дисплей и имеет параметры: номер элемента, с которого начинается вывод, номер последнего выводимого элемента, адрес начала вывода в МОЗУ дисплея.

СТЕРИС чистит часть дисплейного буфера, заполненную операторами рисунка.

РИС запоминает начало рисунка в ДБ и текущие координаты луча.

ЗАМЕЧАНИЕ. Приведенное описание работы подпрограммы не охватывает всего ассортимента выполняемых ими действий, но дает некоторую информацию для понимания механизма трансляции и исполнения.

Примеры трансляции операторов Автокода в операторы ФОРТРАНА.

1. Оператор

МАССИВ A(5) = 1,2,3,4,5, B(5) = 6,7,8,9,10

транслируется в операторы

A = СЧТАТР + I

CALL МАССИВ (A, СЧТЗ, 5)

B = СЧТАТР + I

CALL МАССИВ (B, СЧТЗ, 5)

2. Операторы

ТОЧКА TI (M, N)

ЛУЧ I: TI, (A, B)

транслируется в операторы

TI = СЧТАТР + I

CALL ТОЧКА (TI, 0, M, N)

CALL ЛУЧМ (I, TI)

CALL ЛУЧАМ (I, A, B)

3. Операторы

СВЯЗЬ ФМ1(10), ФМ2(10)

ЛИНИЯ I: ('X'ФМ1, 'Y'ФМ2)

транслируется в оператор

CALL ЛИНИОМ (I, I, ФМ1, 10, ФМ2, 10)

4. Операторы

СТРОКА I (27) = 'ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ'

ТЕКСТ O: I, 'РУССКОГО АЛФАВИТА'

транслируются в операторы

L = СЧТАТР + I

CALL СТРОКА (L, СЧТЗ, 27)

РИСТРО (O, L)

РИСТЕК (O, 'РУССКОГО АЛФАВИТА', 17)

3.4. Реализация процедур Автокода. Описание процедуры Автокода не имеет образа в фортрановской программе. Образом оператора процедуры является совокупность образов всех операторов тела процедуры, в которых формальные параметры текстуально замещены фактическими. По описанию процедуры препроцессор заполняет две рабочие таблицы. Первая из них - КАТАЛОГ - содержит имя процедуры, число ее формальных параметров и ссылку во вторую таблицу - ТАБПРО, которая содержит операторы Автокода и ФОРТРАНА, составляющие тело процедуры. Здесь формальные параметры замещены их внутренним представлением, а именно: вместо каждого параметра стоит признак и номер формального параметра.

Процедуры, использование которых не ограничивается одним запуском задачи, удобно хранить в библиотеке на магнитной ленте (диске). Работа с библиотечными процедурами описана ниже.

3.5. Сервисные возможности системы.

Поиск ошибок и распечатка сообщений стала приведения к ФОРТРАНу. Препроцессор находит ошибки, допущенные в операторах Автокода, проверяя уместность появления символа или лексемы в данном контексте. Простота синтаксиса операторов Автокода и отсутствие вложенности конструкций позволяет практически полностью проверять их синтаксическую корректность непосредственно в процессе узнавания операторов и их параметров. Синтаксис выражений и операторов ФОРТРАНА не проверяется, поскольку результат работы препроцессора обрабатывается транслятором с ФОРТРАНА.

Сообщение об ошибке распечатывается на одной строке АЦПУ вместе с текстом перфокарты, в котором встретилась ошибка. Позиции 1-80 заняты текстом перфокарты, позиции 85-126 - текстом сообщения.

Печать листинга. С помощью управляющей карты * ПЕЧАТЬ, которая ставится перед картой * НАЖЕ, пользователь может заказать печать листинга. Текст каждой перфокарты печатается на одной строке АЦПУ с 1-й по 80-ю позицию. В этом случае со-

сообщения об ошибках печатаются справа от текста перфокарты, в которой обнаружена ошибка в позициях 85-126.

Выдача сообщений этого этапа исполнена и я. На этапе исполнения оттранслированной с ФОРТРАНа программы отыскиваются синтаксические ошибки в операторах Автокода: несоответствие длин двух массивов, по которым задано построение совокупности точек, неверное положение луча и т.п.

В этом случае на экран графического дисплея в его верхний правый угол выводится текст сообщения и номер перфокарты, содержащей оператор Автокода, который вызвал это сообщение. Под номером перфокарты понимается совокупность символов перфокарты, пробитых в позициях с порядковыми номерами от 72 до 80, или порядковый номер перфокарты в тексте, если эта позиция пуста.

Библиотека процедур. Процедуры, использование которых не ограничивается одним запуском задачи, удобно хранить на внешних носителях (дисках или лентах). Управляющая карта БИБЛИОТЕКА МЛ, ЗОНА сообщает системе, что личная библиотека пользователя находится в устройстве с математическим номером МЛ, начиная с зоны ЗОНА.

Описание процедуры с именем вида Б. (имя процедуры) производит запись этой процедуры в личную библиотеку пользователя и стирание предыдущего варианта процедуры с этим именем из личной библиотеки.

Другие сервисные возможности. Поскольку препроцессор формирует задачу для мониторной системы ДУБНА, в распоряжении пользователя находится весь сервис этой системы, а именно: запись задания в архив и редактирование, использование личной библиотеки подпрограмм, комплексация отдельно оттранслированных подпрограмм и т.п.

3.6. Некоторые характеристики реализации Автокода. Так как система состоит из двух функционально различных частей: препроцессора и библиотеки служебных подпрограмм, — то для каждой части системы выбран наиболее подходящий язык реализации. Препроцессор реализован на языке ЯРМО [2], который оказался удобным средством для написания таких программ. Объем ЯРМО-текста, не содержащего сегментов обработки процедур и работы с библиотекой, составляет 9000 слов БЭСМ-6 и транслируется в объектный код объемом 10000 слов БЭСМ-6. Языком реализации биб-

лиотеки подпрограмм послужили ФОРТРАН и автокод МАДЛЕН, работающие в мониторной системе ДУБНА. Объем текста подпрограмм в архиве этой системы составляет примерно 13000 слов и транслируется в объектный код объемом приблизительно 10000 слов БЭСМ-6.

Л и т е р а т у р а

1. Единая система ЭВМ. М., "Статистика", 1974.
2. ГОЛОДОВ В.И., ЧЕБЛАКОВ Б.Г., ЧИНИН Г.Д. Машинно-ориентированный язык высокого уровня для ЭВМ БЭСМ-6. — В кн.: Развитие программного обеспечения БЭСМ-6. М., 1975.

Поступила в ред.-изд.отд.
23 июня 1977 года