

ПС-ЛЯПАС ДЛЯ МАШИНЫ М-20

М.Я. Товштейн

I⁰. Автоматизация составления программирующей
системы

I. В практике программирования задач для УЦМ все большее распространение получают методы автоматического составления программ с помощью вычислительных машин, основанные на использовании алгоритмических языков и программирующих программ (трансляторов). У нас в стране приобрели известность также языки программирования, как операторный [1], адресный [2], АЛГОЛ [3] и соответствующие им трансляторы: ПП-С [4], ПП-УРАЛ-ЗРА [2], ПП-АК [5], ТА-1 [6], ТА-2 [7], система программирования АЛФА [8] и другие.

Автоматизация программирования во много раз повышает эффективность труда программистов, освобождая их от необходимости "натурального" программирования в машинных кодах, делая их работу независимой от конкретной машины и доступной для понимания коллег. Чем больше составляемая программа, тем ощутимей польза от применения методов автоматизации программирования. Однако сами конструкторы трансляторов в своей работе пользуются именно "натуральным" программированием, хотя построение транслятора до определенного уровня можно тоже осуществлять методами автоматического программиро-

вания, как это сделано, например, в компиляторах для языка JOVIAL [9]. Реализация этой идеи для языков типа АЛГОЛ требует вычислительных машин большой мощности и богатого языка, способного описывать логически довольно сложную структуру транслятора.*)

2. В программирующей системе, о которой пойдет речь в этой статье, удалось осуществить принцип автоматизации её построения. Главная заслуга здесь принадлежит, конечно, возможности языка ЛЯПАС [11] представлять алгоритмы решения логических задач. Именно с такими задачами приходится иметь дело при построении программирующей системы (опознавание тех или иных комбинаций символов, составление разного рода таблиц, преобразования над последовательностью символов и т.п.).

В большинстве случаев разработка языка, построение транслятора и конструирование вычислительных машин происходят независимо друг от друга.

При построении трансляторов приходится приспосабливаться к уже существующим типам машин. Из-за этого либо накладывают ограничения на стандартный уровень входного языка, либо при стремлении к тому, чтобы транслятор воспринимал язык как можно полнее, жертвуют качеством получаемой машинной программы. Иногда прибегают к инженерным изменениям в машине [12].

При разработке языка ЛЯПАС, кроме общих требований, предъявляемых к алгоритмическим языкам (наглядность, гибкость, многоступенчатость, однозначность), учитывались еще следующие: наиболее полное использование некоторых особенностей современных вычислительных машин и простота трансляции на машинные языки. Развитие языка ЛЯПАС и соответствующей ему программирующей системы ПС-ЛЯПАС шло параллельно. Это позволило обойти ряд трудностей, возникающих на пути автоматизации построения ПС-ЛЯПАС. Например, в языке фиксировано число переменных, индексов, натуральных констант, за операндами закреплены определенные символы. Эти ограничения в практике программирования не вызывают

*) В работе [10] языком, описывающим алгоритмы трансляции, предлагается сделать КОБОЛ, дополненный некоторыми операторами адресного языка. Однако имеются опасения, что практическое осуществление этой идеи будет весьма затруднительно.

сколько-нибудь серьезных затруднений, а программирующая система от этого только выигрывает. Так, облегчается распознавание операндов в программе, выбор в памяти машины места для хранения их значений можно осуществить заранее. В АЛГОЛе же, например, достаточный произвол в выборе идентификаторов делает необходимым включение в программу описаний, определяющих назначение каждого идентификатора и некоторые свойства связанного с ним объекта [8], а в трансляторе нужно предусматривать анализ идентификаторов и т.п.

3. Построение ПС-ЛЯПАС ведется методом последовательного подключения к системе новых блоков, разрабатываемых с помощью уже существующих. Так, вначале был построен транслятор-1, переводящий выражения с первого уровня ЛЯПАСа на машинный язык. Это был наиболее трудоемкий этап, поскольку программирование алгоритмов трансляции проводилось на машинном языке. Правда, предварительно транслятор-1 был описан с помощью языка ЛЯПАС [13], и это позволило в значительной степени освободить его от ошибок, не прибегая к помощи машины.

Следующие блоки системы программировались уже на ЛЯПАСе. Были изготовлены блоки отладочного режима: корректор, блок подготовки к отладке (подот) и блок выдачи отладочной информации (отинф). С помощью этих блоков проводилась отладка транслятора-2, или компилятора, назначением которого является перевод Л-программ второго уровня в эквивалентные программы, написанные на первом уровне ЛЯПАСа. Теперь уже при составлении других блоков ПС можно использовать второй уровень языка и всю библиотеку подпрограмм, накопленную при эксплуатации настоящего варианта программирующей системы.

Такая методика имеет свои положительные стороны: система всегда находится в эксплуатации, позволяя программистам получать практические результаты, не дожидаясь включения в систему всех блоков; значительно ускоряется и удешевляется процесс отладки блоков программирующей системы, ибо внесение усовершенствований и устранение недочетов в программах производится чрезвычайно просто; имеется возможность, с одной стороны, пользоваться библиотекой подпрограмм, накопленных к данному моменту, с другой стороны — обогащать язык за счет включения в эту библиотеку некоторых фрагментов ПС.

Наиболее сильная сторона ПС-ЛЯПАС заключается в её универсальности. Тот факт, что блоки программирующей системы представляются на алгоритмическом языке, позволяет использовать

её на других машинах почти без изменения. Лишь только один транслятор должен конструироваться заново. Но его объем невелик, и на работу по изготовлению транслятора требуется 2-3 человеко-месяца.

Мы сказали "почти без изменения", так как все же существует незначительная привязка к конкретному представлению информации в ячейке машины. Так, для М-20 содержимое ячейки представляет пять кодов ЛЯПАСа, и в соответствии с этим составлены алгоритмы "распаковки" и "упаковки" Л-программ. Формирование вида отладочной информации также ведется с учетом разрядности машинного слова. В последующем описании мы специально отметим те места в Л-программах блоков ПС, которые должны корректироваться при переходе от одной машины к другой.

2°. Описание блока управления работой системы

1. Программирующая система ПС-ЛЯПАС предназначена для отладки и реализации на машине М-20 программ, написанных на ЛЯПАСе первого и второго уровней. Из таблицы I, в которой приведены некоторые параметры, характеризующие систему, можно заключить, что ПС-ЛЯПАС имеет довольно компактный вид и обладает высокой скоростью формирования машинной программы (порядка 230-280 кодов машинной программы за 1 сек).

Структура ПС изображена на рис. I. Назначение каждого блока может быть понято из его названия. В описываемом варианте системы отсутствуют блоки, которые на рисунке отмечены пунктиром. Поэтому при пользовании настоящим вариантом ПС много еще приходится делать программисту: заботиться об отсутствии синтаксических ошибок, самому выбирать из библиотеки подпрограмм, хранящейся на перфокартах, нужные для решения задачи подпрограммы (оперативную библиотеку) и комплектовать их в определенном порядке, самому распределять место в оперативной памяти (ОЗУ) машины для размещения комплексов и т.д. Но правила, которые нужно помнить программисту, несложны и легко усваиваются.

Данный вариант ПС ориентирован на то, чтобы использовать только ОЗУ машины как при формировании машинной программы (МП), так и при реализации её. Отсюда те ограничения на размеры Л-программ решаемой задачи ($5 \times 585_8$ кодов*) ЛЯПАСа для Л-программ первого уровня и $5 \times 240_8$ кодов для Л-программ вто -

*) Индекс 8 означает восьмеричную систему счисления.

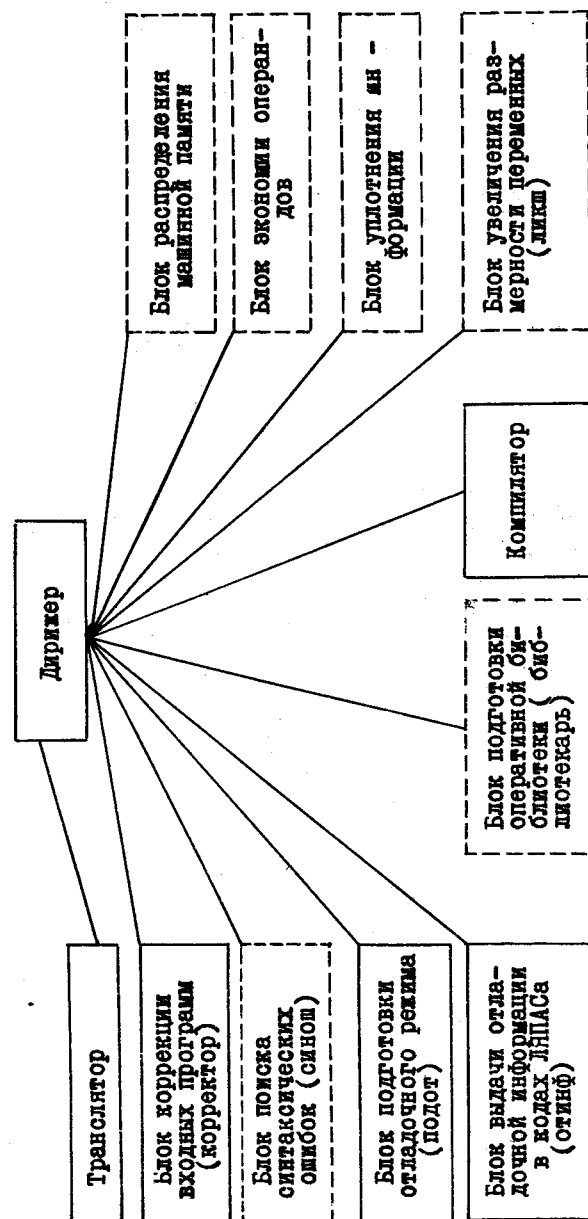


Рис. I. Структура ПС-ЛЯПАС.

Характеристики блоков ПС-ЛЯПАС

Т а б л и ц а 1.

№ п/п	Составные части ПС-ЛЯПАС	Величина программы:			Время ввода с читательного устройства (скорость 200 карт мин)	Время форми- рования машин- ного вида про- граммы
		Число ма- шинных слов	Число слов ЛЯПАСа	Число перфокарт (в т.ч. и непод- ходящих пробитие)		
1.	Транслятор	1437	-	124	38 сек	-
2.	Директор	24+400	684	5+17	7 сек	1,4 сек
3.	Компильтор	700	1058	23	11 сек	4 сек
4.	Подот	234	437	8		
5.	Корректор	205	318	7		
6.	Откиф	118	160	4	1 сек	0,4 сек
	Итого:	3028	2657	178	57 сек	5,8 сек

рого уровня языка), библиотеки подпрограмм (5 x 570₈ кодов) и машинной программы (2621₈ машинных слов), которые указаны в "Инструкции" [14]. В дальнейшем изложении мы предполагаем знакомство читателя с этой статьей.

Построение машинного вида всей системы из Л-программ отдельных блоков проводится по принципу компиляции. Все блоки ПС размещаются в ОЗУ, а о том, какому из них надлежит работать, узнает блок управления системой - **д и р и ж е р** - во время анализа входной информации.

2. Перейдем к рассмотрению программирующей системы в динамике её работы. Для этого нам нужно познакомиться с **программой** **директора**. Этот блок разделен на две части, одна из которых представлена в машинном коде и производит запуск системы, а другая описана на ЛЯПАСе и предназначена для регулирования дальнейшей работы ПС. Команды начального пуска системы вводятся в оперативную память машины в первую очередь. При их исполнении машина вводит с читающего устройства (ЧУ) составные части ПС, формирует с помощью транслятора **машинные программы компилятора, подоты, корректора и директора**.

Транслятор построен так, что может воспринимать Л - программу из любого места ОЗУ и строить машинную программу, предназначенную для работы тоже на любом месте памяти машины. Пользуясь терминами ЛЯПАСа, можно сказать, что транслятор представляет из себя Л - оператор:

транслятор *л ф и з и о н я* ,

у которого операнды, входящие в перечень, имеют следующий смысл:

- л* - указывает начало таблицы, строящейся в процессе трансляции и нужной для присвоения "истинных" адресов командам перехода машинной программы.
 - ф* - указывает "вход" в машинную программу;
 - ш* - указывает начало массива ячеек, где расположена Л-программа, которую нужно перевести на машинный язык;
 - э* - указывает начало поля ОЗУ, на котором формируется машинная программа (МП) во время трансляции (в "условных" адресах);
 - ю* - указывает начало поля, где должна располагаться МП при реализации (в "истинных" адресах);
 - я* - указывает длину сформированной машинной программы.
- (Заметим, что этими операндами служат соответствующие ин-

дексы, взятые из дополнительного набора операндов ЛЯПАСа).

Такая организация транслятора позволяет машинную программу дирижера разместить в старших адресах ОЗУ машины ($7135 \div 7757$), откуда ему легче управлять другими блоками системы, расположенными в ячейках $3400 \div 5565$. Сам транслятор занимает массив ячеек с 1000 по 3356, из них первые 2170 ячеек содержат "рабочую часть" транслятора *).

После трансляции на машинный язык Л-программ компилятора, подота и корректора в память машины автоматически вводится Л-программа отинфа. Теперь ПС-ЛЯПАС готов к работе.

Из таблицы I видно, что на подготовку ПС к работе тратится около 60 сек, из них львиная доля (92%) уходит на ввод перфокарт с ЧУ. Поэтому целесообразнее хранить систему в виде, подготовленном к обработке входной Л-программы, на магнитном барабане (или ленте), автоматически считывая её (на это уходит 0,7 сек) в оперативную память после решения очередной задачи. Запись программирующей системы на барабан ** и восстановление её в ОЗУ также поручено делать той части программы дирижера, которая представлена машинными командами.

Использование магнитного барабана можно блокировать. Для этого из комплекта перфокарт с ПС-ЛЯПАС вынимается перфокарта № 4. Она содержит снабженные адресными кодами команды, отсутствие которых даёт возможность дирижеру не выполнять операции обмена информацией между ОЗУ и внешней памятью.

3. После того, как дирижер подготовит ПС к работе (и запомнит её на барабане), он вводит в машину Л-программу решаемой задачи вместе с кодами $k_2 \div k_n$, несущими информацию о необходимости использования того или иного блока ПС, о режиме отладки, об исходных данных задачи и переходит к исполнению второй части своей программы.

Эта часть программы дирижера написана на первом уровне языка ЛЯПАС, поскольку предназначена для непосредственного восприятия транслятором. Но для наглядности представления её структуры запишем её в виде программы второго уровня:

*) Номера ячеек мы будем давать в восьмеричной системе счисления.

**) Настоящий вариант ПС-ЛЯПАС использует магнитный барабан № 2. Выбор именно этого барабана продиктован тем, что иногда другие барабаны бывают заняты стандартными программами ИС-2.

управление $k_2 \div k_n$ // страхикл AB //
припотинф AB // подтранс AB лфшэюэ //
обабисх $k_4 k_5 k_6 ABGH$ // $\dagger \varphi$.

Поясним назначение каждого из входящих в программу дирижера операторов.

Оператор управление приводит коды $k_2 \div k_n$ к рабочему виду и, анализируя значение k_2 , определяет, какой блок ПС должен работать. Здесь необходимо отметить, что формирование МП дирижера и программы других блоков проходит независимо друг от друга. Связь между двумя самостоятельными машинными программами, полученными при трансляции, можно осуществить с помощью операций $\dagger \psi$ и $\delta \psi$ [15]. Операциями локализации $\delta z, \delta y, \delta x$ начинаются Л-программы компилятора, подота и корректора, соответственно, а переход к ним происходит посредством операции $\dagger \varphi$, где $\varphi \in \{x, y, z\}$.

Заметим, что обрабатываемая программа каждым блоком воспринимается как комплекс A , а получающаяся в результате - представляется комплексом B , причем значения элементов этих комплексов имеют вид 45-разрядных кодов. В соответствии с тем, какой блок ПС должен работать, оператор управление присваивает элементам α_0 и $\alpha_i \in A$ определенные значения, задающие место в ОЗУ для упомянутых комплексов A и B .

Операторы страхикл и припотинф работают в том случае, если входная Л-программа должна реализоваться в режиме отладки. Дело в том, что в программе, подлежащей отладке, может оказаться ошибка, которая приведет к повторению какого-либо участка программы очень большое число раз (как говорят, произойдет закикливание). Чтобы избежать излишней траты машинного времени и дать возможность ПС обрабатывать другие программы, при реализации данной Л-программы предусматривается "страхование от закикливания" - подсчет числа обращений к началу каждого предложения. Как только это число достигнет определенного порога, решение данной задачи прекращается и вводится Л-программа следующей.

Такой подсчет ведется в § 177 блока отинф (см. далее), приписываемого оператором припотинф к программе решаемой задачи, а обращение к счетчику из этой программы осуществляется благодаря тому, что страхикл вставляет в неё после символов δ ($\delta \in \{0, 1, \dots, 170\}$) символы $\leftrightarrow$ 177. Таким образом "застрахованная от закикливания" Л-программа транслируется на машинный язык.

Формирование машинной программы проводится с помощью оператора подтранс. Этот оператор вводит с ЧУ значения специаль-

№ п/п	Сим- вол опе- ран- да	Код	Значение операнда	Представление значения операнда при вводе в машину М-20	Роль операнда при работе дирижера (ЛП-Л-программа)
I.	$\underline{a}$	240	3400	000 0000 0007 0000	Задаёт стандартное начало поля ОЗУ, на кото- ром располагается МП при трансляции.
2.	$\underline{b}$	241	2720	000 0000 0005 6400	Задаёт "вход" в транслятор.
3.	$\underline{c}$	242	6400	000 0000 0015 0000	Указывает место, откуда считывается ЛП для коррекции и подготовки к отладке.
4.	$\underline{d}$	243	5600	000 0000 0013 4000	Указывает место, куда записывается скорректи- рованная ЛП при работе корректора.
5.	$\underline{f}$	245	5800	000 0000 0012 6000	Указывает место, куда записывается результат работы блока подот.
6.	$\underline{q}$	246	6225	000 0000 0014 4520	Указывает место, куда записывается ЛП, кото- рый должна восприниматься транслятором.
7.	$\underline{h}$	247	5160	000 0000 0012 3400	Задаёт начало библиотеки подпрограмм.
8.	$\underline{z}$	250	5750	000 0000 0013 7200	Задаёт начало одного из рабочих комплексов компьютера.
9.	$\underline{q}$	257	5224	000 0000 0014 4500	Указывает начало таблицы команд перехода для транслятора.
10.	$\underline{z}$	260	241	000 0000 0000 5020	Задаёт размер поля, на котором размещается ЛП второго уровня.
11.	$\underline{s}$	261	714	000 0000 0001 6300	Задаёт размер поля, на котором размещается ЛП, воспринимаемая транслятором.
12.	$\underline{x}$	371	7505	000 0000 0017 2120	Задаёт начало рабочего поля блока отлад.

ных комплексов A и B , придаёт им "рабочий" вид и обращает-
ся к транслятору, предварительно напечатав номер, под которым
Л-программа поступила в ПС. После работы транслятора печатает-
ся число, указывающее длину получившейся МП, а сама машинная
программа перемещается, если это необходимо, на поле, указан-
ное значением дополнительного индекса $ю$. Если программист
предусматривает выдачу МП на печать, оператор подтранс ана-
лизирует число команд МП и, если оно превышает 511, печатает
первые 510. Это связано с тем, что для вывода информации в ма-
шине М-20 используется буферный регистр емкостью 511 кодов. При
выдаче большего числа кодов первые коды выводимого массива мо-
гут быть испорчены. Напечатанная МП представляет интерес, в
основном, для конструкторов ПС, и 510 кодов обычно бывает до-
статочно.

Оператор обработкс вводит в память машины исходные данные
для решения задачи и в соответствии с кодами K_4 , K_5 и K_6
обрабатывает переменные, индексы и комплексы, сдвигая их зна-
чения на четыре позиции влево. Затем начинается реализация
транслированной МП. Этот момент отмечается печатанием номера
решаемой задачи.

После реализации МП выполняется заключительная часть про-
граммы дирижера, представленная машинными командами. В счет-
чик числа обработанных Л-программ (ячейка 0013) добавляется
единица, и значение его сравнивается с содержимым ячейки 0001,
где программист заранее указывает количество подлежащих реше-
нию задач. Затем либо происходит останов машины в ячейке 0663,
либо ПС восстанавливается в ОЗУ, если она была сохранена на
барабанах, и вводится Л-программа следующей задачи. Таким об-
разом, ПС-ЛЯПАС, будучи один раз введенной в машину, позволя-
ет обработать серию Л-программ.

4. Ниже приводится программа дирижера, написанная на язы-
ке ЛЯПАС первого уровня. Во время работы этой программы исполь-
зуются операнды, назначение которых видно из таблицы 2. Опе-
ранды, над которыми стоит тильда ($\sim$), взяты из дополнитель-
ного набора.

управление

$b_0 < 4 \Rightarrow b_0 \cdot 2 \Rightarrow a + 10 \Rightarrow e$ Приведение кодов $K_2 \div K_n$ к рабо-
 $\leftrightarrow 30 \subseteq \Rightarrow a_0 \Rightarrow u$ чему виду.

$k_2 \wedge c_0 \cdot 1 d \Rightarrow a_1 c_{27} + c_{37} \Rightarrow a_2 + 1$ Обращение к корректору.

$d \Rightarrow a_0 b_1 \Rightarrow b_0 \subseteq \leftrightarrow 20 \subseteq \Rightarrow a_0$ Перепись скорректированной Л-прог-
раммы (ЛП).

§ 1 $q \Rightarrow \bar{h} k_2 \wedge c_2 \Rightarrow 2 f \Rightarrow a_1 \wedge a_7 \Rightarrow a_0 b_1 \Rightarrow b_1$	Обращение к подоту.
§ 2 $k_2 \wedge c_3 \Rightarrow 6 \quad q \Rightarrow \bar{u} \Rightarrow a_1$ $b_0 - \bar{u} \Rightarrow 3 \quad a_1 * \dagger$	Подготовка к компиляции. Проверка: входит ли головная программа в отведенное ей место
§ 42 $\dagger$ § 43 $\dagger \rightarrow 41$	Аварийные выходы.
§ 3 $\bar{h} - b_0 \Rightarrow c \leftrightarrow 20 \quad c \Rightarrow a_0$ $o h_0 \Rightarrow h_1 \Rightarrow h_2 \Rightarrow h_3$ $\Rightarrow h_4 \Rightarrow h_5 \Rightarrow h_6 \Rightarrow h_7$ $\dagger 177 \Rightarrow q \quad \bar{c} \Rightarrow a_2$ $k_2 \wedge c_2 \Rightarrow 4 \quad 171 \Rightarrow q$	Перепись головной программы. Сигнал компилятору об отсутствии "штриховки". Ввод библиотеки. Указание компилятору номера последнего допустимого предложения.
§ 4 $\bar{x} - 2 \Rightarrow \bar{x} \quad \dagger \bar{x} \quad n \rightarrow 42$ $a_1 \Rightarrow a_0 b_1 \Rightarrow b_0 \quad q \oplus 171 \rightarrow 5$ $\bar{a} \leftrightarrow 20 \quad \bar{a} \Rightarrow a_0$	Обращение к компилятору. Перепись скомпилированной программы.
§ 5 $k_2 \wedge c_{22} \Rightarrow 6 \leftrightarrow 26 \quad \underline{A} \bullet \bullet$	Выдача скомпилированной программы.

страхцикл

§ 6 $k_2 \wedge c_2 \Rightarrow 22 \quad q \Rightarrow \bar{u} \Rightarrow a_1 \bar{p} j \bar{o} k \rightarrow 11$	Опознавание символов., $\bar{z}, \leftarrow$ Перепись кодов, следующих за $\leftarrow$.
§ 7 $\leftrightarrow 16 \oplus 6 \Rightarrow 13 \bar{b} \oplus 1 \Rightarrow 12 \bar{b} \oplus 7 \rightarrow 7 \rightarrow 16$	
§ 11 $\leftrightarrow 16 \leftrightarrow 16 \leftrightarrow 16 \rightarrow 7$	
§ 12 $\leftrightarrow 16 \quad 45 \leftrightarrow 17 \quad c_{27} \wedge 177 \leftrightarrow 17 \rightarrow 7$	Запись символов $\leftrightarrow 177$.

припотинф

§ 13 $\bar{\Delta} k \quad c_{27} \Rightarrow a_0 \bar{p} j$	Отинф представляется как комплекс $\underline{A}$.
§ 14 $\bar{Q} - \bar{s} \Rightarrow 15 \quad a_4 * \rightarrow 42$	Проверка: входит ли ЛП в отводимое для неё место
§ 15 $\leftrightarrow 16 \oplus 6 \rightarrow 14 \rightarrow 22$	Считывание и запись ЛП. Это место программы должно быть скорректировано при переходе к другой машине.
§ 16 $\Delta j: 5 \times 11 \Rightarrow p \quad a_2 \neq p \wedge f_7$	
§ 17 $\Rightarrow \bar{b} \Delta k: 5 \times 11 \Rightarrow p \quad 100 - p \Rightarrow q$ $b \neq q \Rightarrow p f_7 \neq q \oplus f_1 \wedge b_2 \vee p \Rightarrow b_2$ $b!$	

подтранс

§ 20 $\Rightarrow a \bar{o} b$	Программа для переписи ЛП или МП с одного места на другое.
§ 21 $\bar{a} b \Rightarrow k_a \Delta a \Delta b - b_0 \Rightarrow 21!$	
§ 22 $\bar{a} \Rightarrow \bar{b} \Rightarrow \bar{f} \bar{o} \dagger a \oplus \bar{f} \bar{o} \Rightarrow 23 \bar{f} \bar{o} \wedge 4 \Rightarrow \bar{f} \bar{o}$	
§ 23 $100 \Rightarrow a + 40 \Rightarrow e \leftrightarrow 30$ $f_1 \Rightarrow a \quad f_2 \Rightarrow e \leftrightarrow 30$ $k_{13} * e_4 \Rightarrow k_{12} \quad \dagger \bar{b} \quad n \rightarrow 43$ $a_0 \Rightarrow \bar{a} b_0 \Rightarrow \bar{w} \quad \bar{x} \Rightarrow b_0 *$ $\bar{g} \Rightarrow a_0 \oplus \bar{f} \bar{o} \Rightarrow 24 \bar{f} \bar{o} \leftrightarrow 20$	Ввод комплексов A, B Приведение комплексов A и B к рабочему виду. Обращение к транслятору. Перепись МП на другое место.
§ 24 $k_2 \wedge c_{24} \Rightarrow 27 \bar{x} - f_6 \Rightarrow 25 f_5 \Rightarrow b_0$	Выдача на печать машинной программы.
§ 25 $\leftrightarrow 26 \rightarrow 27$	
§ 26 $\underline{A} \bullet!$	

обрабисх

§ 30 $k_a \neq 44 \Rightarrow k_a \Delta a - e \Rightarrow 30!$	Ввод исходных данных.
§ 27 $\bar{u} \Rightarrow a_0 \quad \bar{w} \Rightarrow b_0$	
§ 31 $\dagger \bar{o} \bar{b} \bar{o} d a \Rightarrow f$	
§ 32 $\bar{o} a \Delta d - 40 \rightarrow 33 \quad k_6 \wedge c_a \Rightarrow 32$ $a_a \Rightarrow a + b_a \Rightarrow e \leftrightarrow 30 \rightarrow 32$	Сдвиг значений комплексов на 4 позиции влево.
§ 33 $\Delta a - 40 \rightarrow 34 \quad k_4 \wedge c_a \Rightarrow 33 q_a \leftarrow 4 \rightarrow q_a \rightarrow 33$	Сдвиг значений переменных.
§ 34 $f \Rightarrow a \bar{o} b$	Сдвиг значений индексов.
§ 35 $\Delta b - 40 \rightarrow 36 \quad k_5 \wedge c_8 \Rightarrow 35 h_f \leftarrow 4 \rightarrow h_f \rightarrow 35$	
§ 36 $k_2 \wedge c_2 \Rightarrow 40 \quad o k_{14} o l 40 \Rightarrow b_{32} \Rightarrow b_{33}$ $\bar{x} \Rightarrow a_{32} + 40 \Rightarrow a_{33} \quad o b$	Подготовка исходных данных для отинфа.
§ 37 $o \underline{a} b \Delta b \oplus 40 \rightarrow 37$	

§ 40 $k_{13} * d_4 \Rightarrow k_{12} a \Rightarrow a^* \varphi$] Переход к реализации МП.

$k_2 \wedge c_{10} \rightarrow 31$

§ 41.

Проверка: имеет ли место ре-
жим решения задачи с меняющим-
ся набором входных значений
операндов?

5. Как уже упоминалось, дирижер перед работой очередного блока ПС указывает, откуда этот блок должен воспринимать Л-программу и куда помещать результат своей работы. Условимся ре-
жим работы ПС (коррекция, отладка, компиляция, коррекция с компиляцией и т.п.) делить на этапы, при реализации которых работает только один блок системы. Каждый режим завершается этапом, когда строится машинная программа, если не считать работы блока ввода и обработки исходных данных. На рис.2, который дает представление о распределении ОЗУ машины М-20 во время работы системы, этот этап отражен только для первого режима, так как для других режимов распределение памяти на этом этапе точно такое же. Обращаем внимание и на то, что для большинства этапов на рисунке изображаются только те участки памяти, на которых происходит перемещение информационного материала. Работающий блок отмечен в соответствующей графе цифрой "1".

3°. Блоки отладочного режима

1. При использовании систем автоматического программирования большое значение имеет методика обнаружения и устранения ошибок в программах, написанных на алгоритмическом языке. Если программист не имеет информации о ходе решения задачи в той форме, в какой он записывал программу, ему нужно, кроме алгоритмического языка, знать и язык машины, а это сводит на нет преимущества программирующих систем.

В ПС-ЛЯПАС вопросам отладки Л-программ уделено особое внимание. Как и во многих других системах автоматического программирования, здесь предусмотрен поиск в программах так называемых синтаксических ошибок. Блок сином [16] выдает программисту координату символа, наличие которого создает недопустимую в языке комбинацию символов, и вид этой комбинации. Если у программиста возникает необходимость внести исправления в программу, он задает информацию блоку коррекции: границы изменя-

*) С помощью операции $a \Rightarrow a$ происходит занесение на адресный регистр М-20 исходного значения индекса a , которому сопоставляется этот регистр.

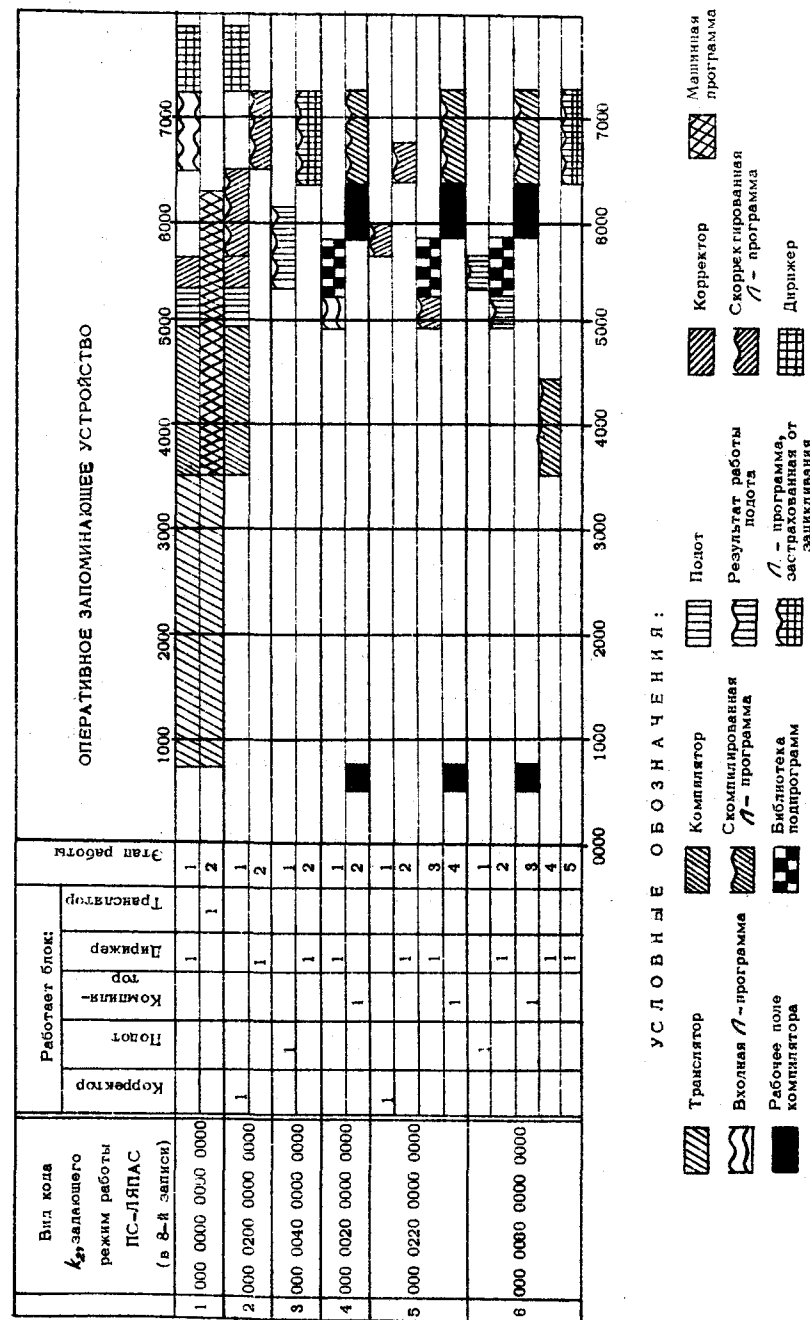


РИС. 2. РАСПРЕДЕЛЕНИЕ ОЗУ.

ющих частей Л-программы и коды символов, которые следует вставить в программу. Предусматривается выдача отладочной информации в процессе решения задачи. При этом существенным обстоятельством является то, что она кодируется в виде, максимально приближающемся к кодам символов языка ЛЯПАС. Таким образом, программист держит связь с машиной на уровне входного языка.

Рассмотрим программы тех блоков отладочного режима, которые присутствуют в настоящем варианте ПС-ЛЯПАС: программы корректора, отинфа и подота.

2. Блок коррекции Л-программ - корректор - предназначен для того, чтобы автоматизировать процесс внесения исправлений в программу, освободив программиста от необходимости "сдвигать" и "раздвигать" её, вновь кодировать и переносить на перфокарты. На основании ИНКОРа [14] всю нужную работу выполняет корректор, выдавая программисту напечатанный и отперфорированный текст исправленной программы.

При реализации алгоритма коррекции Л-программ предполагается, что исходная программа представляется комплексом A, исправленный её вариант - комплексом B, ИНКОР - комплексом C. Место расположения этих комплексов в ОЗУ машины указывается дирижером. Операнды в программе корректора взяты из специального комплекса J. Индексы j, c и k служат счетчиками символов, соответственно, в ИНКОРе, исходной и скорректированной Л-программе. Необходимо отметить, что код 000 Л-программы рассматривается наравне с остальными.

Л-программа корректора имеет следующий вид *):

§ 134 $\bar{0} \bar{x} \bar{0} j \rightarrow 154 \leftarrow 155$ $\leftarrow 154 \leftarrow 155 \leftarrow 154 \leftarrow 155$	Перенос кодов напки Л-программы.
§ 135 $\leftarrow 153 \oplus 36 \rightarrow 156 \vee \oplus 37 \rightarrow 137$ $400 \oplus \bar{b} + j \Rightarrow f$	Опознавание символов $\oplus$ и $\odot$ в ИНКОРе.
§ 136 $j \oplus f \rightarrow 135 \leftarrow 153 \leftarrow 155 \rightarrow 136$	Исходное состояние счетчика символов Л-программы.
§ 137 $2 \Rightarrow c \leftarrow 153 \Rightarrow u$	
§ 140 $\leftarrow 154 \oplus 7 \rightarrow 141 \ c + 4 \Rightarrow c \rightarrow 140$	Перебор кодов, стоящих за операторами $\leftarrow$ и $\leftarrow$.
§ 141 $\bar{a} \oplus 34 \rightarrow 142 \ \Delta c \rightarrow 140$	

*) Здесь и в программе подот использованы "старшие" номера предложений, поскольку "младшие" заняты в компиляторе.

§ I42 $d \oplus 1 \rightarrow 140 \rightarrow 154 \oplus 1 \rightarrow 140 \rightarrow 153$ $c \Rightarrow v + b - 2 \Rightarrow c \rightarrow 153 \oplus 1 \rightarrow 151 \rightarrow h$	Проверка: располагается ли неизменяемая часть Л-программы в пределах одного предложения.
§ I43 $\rightarrow 154 h \rightarrow 150 d \oplus 7 \rightarrow 145$ $d \rightarrow 155 \bar{0} z$	Перепись кодов, относящихся к оператору $\leftarrow$.
§ I44 $\Delta z \oplus 4 \rightarrow 143 \rightarrow 154 \rightarrow 155 \rightarrow 144$	
§ I45 $d \oplus 34 \rightarrow 146 d \rightarrow 155 \rightarrow 154 \rightarrow 155 \rightarrow 143$	Перепись номера Л-оператора.
§ I46 $d \oplus 1$	
§ I47 $\Rightarrow h d \rightarrow 155 \rightarrow 143$	Перепись кодов из комплекса А в комплекс В до символа предложения, указанного в позиции правой границы.
§ I50 $d \oplus b \rightarrow 147 c \Rightarrow v - 1 \Rightarrow c$	
§ I51 $\rightarrow 153 + v \Rightarrow v$	
§ I52 $c + 1 \oplus v \rightarrow 135 \rightarrow 154 \rightarrow 155 \rightarrow 152$	Перепись следующих кодов.
§ I53 $\Delta j : 5 \times 11 \Rightarrow p \underline{c}_2 \neq p \wedge f_7 \Rightarrow b!$	Считывание кодов ИНКОРа.
§ I54 $\Delta c : 5 \times 11 \Rightarrow q \underline{a}_2 \neq q \wedge f_7 \Rightarrow d!$	Считывание кодов корректуры Л-программы.
§ I55 $\Rightarrow e \Delta k : 5 \times 11 \Rightarrow q 100 - q \Rightarrow z$ $e = z \Rightarrow q f_7 \neq z \oplus f_7 \wedge b_2 \vee q \Rightarrow b_2!$	Запись кодов в исправленную программу.
§ I56 $k : 5 \quad 1 + 2 \Rightarrow b, \underline{B} \oplus \underline{B} \oplus \underline{B}$	Выдача скорректированной программы.
$10 > 10 \Rightarrow 10 \uparrow 10.$	Выход из корректора в дирижер.

Если пользоваться этой программой на другой машине, нужно в § I53, § I54 и § I55 изменить константы при вычислении величины сдвига машинного кода, в § I56 при нахождении мощности комплекса $\underline{B}$ значение k нужно делить на число "упаковываемых" в ячейку кодов ЛЯПАСа. Процедура выхода из программы корректора существенным образом зависит от операции выхода в язык машины, которая (операция) связана с конкретным представлением в машине команды передачи управления. Это тоже надо учитывать при применении данного алгоритма на другой машине.

3. При наличии отладочного режима происходит такое преобразование входной программы, чтобы во время её реализации выдавалась информация, позволяющая следить за изменениями значений переменных и индексов. В этом варианте ПС печатаются значения только тех операндов, которые стоят после операторов Δ , $\bar{\Delta}$, $\circ$, $\bar{\circ}$, $\Rightarrow$, $\Leftarrow$, $\neq$, $\neq$. О том, какие именно операнды интересуют программиста, ПС узнает по значениям кодов K_7

и K_{10} [I4]. Блок подготовки Л-программы к отладке - подот - вставляет в определенных точках входной программы операторы обращения к блоку отинф, который работает во время реализации машинной программы и сопровождает выдачу значений операндов индексацией, позволяющей узнать код операнда и номер предложения Л-программы, где этот операнд принимает текущее значение.

Перейдем к рассмотрению алгоритма работы блока подот.

Анализируя Л-программу второго уровня, подот находит прямую форму упоминания Л-оператора с номером N и, если 34-й разряд кода K_7 отмечен единичным значением, после символа $//$ ставит выражение $\Rightarrow 000\ 000\ 000\ N\ 34 \rightarrow 174$.

Для предложений, номера которых входят в "отрезок", задаваемый кодом K_H [I4], после символов $\S\ \delta$ ($\delta \in \{0, 1, \dots, 171\}$) записывается выражение $\delta \Rightarrow \underline{ш} \underline{э}$, обеспечивающее фиксирование номера δ предложения индексом $\underline{ш}$ и запоминание значения собственной переменной $\underline{э}$ Л-программы индексом $\underline{э}$. Кроме этого, если переменная, скажем, $\underline{b}$, или индекс $\underline{d}$ отмечены соответственно в кодах K_7 и K_{10} , то после операций присвоения этим операндам нового значения ставится выражение $\Rightarrow \underline{э}\ 01 \rightarrow 174$ - для переменной $\underline{b}$ и $\Rightarrow \underline{э}\ 43 \rightarrow 173$ - для индекса $\underline{d}$. В результате индекс $\underline{э}$ представляет значение операнда, а значение собственной переменной $\underline{э}$ позволит блоку отинф отметить это значение соответствующим образом.

При разборе программы блока подот надо иметь в виду, что комплекс $\underline{A}$ представляет обрабатываемую программу, индекс $\underline{/}$ - счетчик символов в ней, комплекс $\underline{B}$ - обработанную программу, индекс $\underline{k}$ - счетчик символов в ней. Значения индексов $\underline{s}$, $\underline{t}$ и $\underline{u}$ представляют соответственно номера "левого" и "правого" предложений "отрезка" и номер анализируемого предложения Л-программы. Все операнды программы подот взяты из дополнительного набора индексов. Напомним, что коды K_7 , K_{10} и K_H к началу работы подота имеют "рабочий" вид.

Блок подготовки Л-программы к отладке работает по следующей программе:

$\delta \underline{y} \bar{\delta} j \bar{\delta} k\ K_H \neq 16 \vee c_{27} \vee c_{16} \Rightarrow s \wedge f_7 \Rightarrow t$ $s > 11 \Rightarrow s \rightarrow 162$	Выделение границ "отрезка".
§ I60 $\rightarrow 174 \oplus 6 \rightarrow 177 \bar{b} \oplus 1 \rightarrow 171$ $\bar{b} \oplus 55 \rightarrow 160 \bar{b} \wedge f_6 \Rightarrow i \oplus 20 \rightarrow 161$ $\rightarrow 174 \rightarrow 160$	Опознавание символов $\circ, \S, +, ', ', ', ', ''$.

Как следует из приведенной программы, при формировании вида выводимой отладочной информации используется 45 разрядов машинного слова М-20. Например, если в § 17 переменная d приняла значение

110011000000000000000000000011.

то отинф образует код (в восьмеричной записи):

170381400000003

Здесь подчеркнуты: номер предложения, отметка операнда и его значение.

Таким образом, программа блока отинф при использовании ПС-ЛЯПАС на другой машине должна подвергнуться значительным изменениям, однако алгоритм его работы применим и в этом случае.

4°. О трансляторах ПС

Мы привели программы работ блоков отладочного режима и блока управления системой. Описания трансляторов - со второго уровня языка на первый и с первого на машинный язык - дадим лишь в основных чертах (достаточно подробно они описаны в [17]).

Транслятор-2, или компилятор, на основании Л-программы второго уровня и оперативной библиотеки подпрограмм монтирует новую программу. На место Л-оператора и перечня его внешних операндов подставляется соответствующая подпрограмма, настроенная на работу с указанным перечнем. Если во вставляемой подпрограмме тоже есть какие-либо Л-операторы, то в ней, в свою очередь, вставляются соответствующие им подпрограммы и т.д. При этом исходная программа "раздвигается" и в ней вводится единая нумерация предложений. Вообще говоря, получаемая на выходе компилятора программа может содержать операции над сложными переменными - компилятор производит "штриховку" переменных на основе поданной программистом информации. Для данного варианта ПС дирижер сигнализирует компилятору, что установку штрихов делать не следует. Поэтому скомпилированная программа может сразу обрабатываться транслятором-1.

Этот транслятор воспринимает код оператора первого уровня ЛЯПАСа как имя некоторой машинной программы (которая может состоять из одной или нескольких команд), способной реализовать данный оператор. Анализируя оператор, транслятор выбирает из заранее составленной совокупности таких программ ту, что носит данное имя. В результате обработки всей Л-программы образуется последовательность программ, которые выполняют те же функции и в том же порядке, что и операторы Л-программы. Эту по-

следовательность программ мы и называем машинной программой.

Анализ получаемых с помощью ПС-ЛЯПАС программ позволяет сказать, что их быстроедействие и размеры довольно близко подходят к аналогичным показателям для программы, составленных квалифицированным программистом.

5°. Заключение

Программирующая программа на основе языка ЛЯПАС появилась на свет осенью 1963 года. Постепенно совершенствуясь, ЛЯПАС и соответствующий ему транслятор меняли свой вид. Тот вариант ПС-ЛЯПАС, который описан в этой статье, эксплуатируется с июля 1965 года. С помощью ПС было отлажено около 300 Л-программ различной сложности, характеризуемой, в какой-то мере, длиной соответствующих машинных программ - от 300₈ команд до 2760₈. Во время эксплуатации системы высказывались ценные замечания по улучшению некоторых её сторон. Многие предложения уже учтены, кое-что может быть реализовано в следующих вариантах ПС (например, возможность следить за изменениями значений операндов не только в головной программе, но и в её подпрограммах - до некоторой глубины; автоматический "перевод латинских букв в греческие", и наоборот и т.п.)

Пользуясь случаем, автор приносит глубокую благодарность руководителю работы, кандидату физико-математических наук старшему научному сотруднику А.Д. Закревскому, которому принадлежит большинство идей, положенных в основу ПС-ЛЯПАС. Автор благодарит ст. инженера В.Г. Новоселова, аспирантов В.И. Островского и Н.Р. Торопова за полезные советы, способствовавшие усовершенствованию системы, а также сотрудников лаборатории счетно-решающих устройств Сибирского физико-технического института, которые прошли через все стадии готовности ПС к использованию и не потеряли при этом оптимизма.

Работы, связанные с вычислительной машиной, проводились в Вычислительном центре СО АН СССР. Автор признателен всем сотрудникам ВЦ за ту помощь, которую ему оказали при отладке и эксплуатации программирующей системы.

ЛИТЕРАТУРА

1. А.А. Ляпунов. О логических схемах программ.- Проблемы кибернетики, М., Физматгиз, 1958, вып. I, стр. 46-74.
2. Е.Л. Ющенко. Адресное программирование. Киев, Гостехиздат УССР, 1963.
3. С.С. Лавров. Универсальный язык программирования (АЛГОЛ-60), М., "Наука", 1964.
4. А.И. Китов, Н.А. Криничский. Электронные вычислительные машины и программирование. М., Физматгиз, 1961.
5. В.И. Глушков, Е.Л. Ющенко. Вычислительная машина "Киев". Киев, Гостехиздат УССР, 1962.
6. В.Н. Попов, В.А. Степанов, А.Г. Стишева, Н.А. Травникова. Программирующая программа. - журн. вычислит. математики и матем. физики, 1964, 4, №1, стр. 78-95.
7. М.Р. Шура-Бура, Э.З. Любимский. Транслятор АЛГОЛ-60. - Журн. вычислит. математики и матем. физики, 1964, 4, №1, стр. 96-112.
8. Альфа - система автоматизации программирования. - Сб. статей под ред. А.П. Ершова. Новосибирск, Редакционно-издательский отдел СО АН СССР, 1965.
9. J.I. Schwartz. JOVIAL - a general algorithmic language. Symbolic languages in data processing. Proc. of the Symp., ICC, Rome, 1962, Gordon and Breach Science Publ., New York-London, 1962, p. 481-493.
10. Л.П. Бабенко. Об использовании языка типа КОБОЛ для описания трансляторов. - Кибернетика, 1965, I, № 5, стр. 41-45.
11. А.Д. Закревский. Описание языка ЛЯПАС. - Логический язык для представления алгоритмов синтеза релейных устройств, М., "Наука", 1966, стр. 7-38.
12. М.Р. Шура-Бура, В.В. Мартынюк. Об эффективной организации динамического использования памяти. - Журн. вычислит. математики и матем. физики, 1964, 4, № 5, стр. 963-967.
13. М.Я. Товштейн. Транслятор для быстродействующих УЦВМ. - Логический язык для представления алгоритмов синтеза релейных устройств, М., "Наука", 1966, стр. 52-69.

14. М.Я. Товштейн. Инструкции к программирующей системе ЛЯПАС. - Данный сборник, стр. 87-116.
15. А.Д. Закревский. Дополнительные операторы первого уровня ЛЯПАСа. - Данный сборник, стр. 85-86.
16. Н.А. Усачева. Поиск синтаксических ошибок в Л-программах. - Логический язык для представления алгоритмов синтеза релейных устройств, М., "Наука", 1966, стр. 75-78.
17. Логический язык для представления алгоритмов синтеза релейных устройств. М., "Наука", 1966.

Сибирский физико-технический
институт

Поступила в редакцию
20. III. 1966 г.