

Ц 8419  
К-238

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

ДУБНА

2/1-78



121/2-78

11 - 10967

А.А.Карлов, А.Д.Полынцев

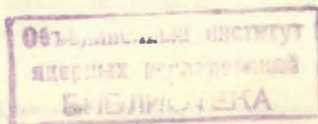
ЛОКАЛЬНОЕ МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
УДАЛЕННОЙ ДИСПЛЕЙНОЙ СТАНЦИИ

1977

11 - 10967

А.А.Карлов, А.Д.Полынецв

ЛОКАЛЬНОЕ МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
УДАЛЕННОЙ ДИСПЛЕЙНОЙ СТАНЦИИ



Карлов А.А., Полинцев А.Д.

11 - 10967

Локальное математическое обеспечение удаленной дисплейной станции

Описывается локальное математическое обеспечение удаленной дисплейной станции на базе малой ЭВМ М-6000, подключенной к центральной ЭВМ БЭСМ-6 через канал связи.

Приводится описание специального математического обеспечения, предназначенного для организации локального управления работой дисплейной станции в соответствии с командами, поступающими от центральной ЭВМ и от человека.

Предлагается метод реализации асинхронного управления для дисплейной станции, который может найти применение при разработке математического обеспечения в системах передачи данных.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1977

## I. Введение

Математическое обеспечение удаленной дисплейной станции (УДС) на базе ЭВМ М-6000, подключенной через канал связи к центральной ЭВМ БЭСМ-6 измерительно-вычислительного комплекса ОИЯИ, состоит из комплекса подпрограмм для УДС на ЭВМ БЭСМ-6 и локального математического обеспечения на ЭВМ М-6000 /1,2/.

В данной работе рассматривается локальное математическое обеспечение (МО) с точки зрения его внутренней структуры.

Локальное математическое обеспечение можно разделить на две части:

- общее математическое обеспечение;
- специальное математическое обеспечение.

Общее математическое обеспечение состоит из стандартного (т.е. поставляемого вместе с машиной) МО ЭВМ М-6000 и математического обеспечения графического дисплея СИГДА.

Из стандартного МО в настоящее время используются драйверы внешних устройств ввода-вывода ЭВМ М-6000, подпрограмма управления вводом-выводом и перемещающий загрузчик (т.е. некоторая конфигурация основной управляющей системы - ОУС), а также ряд библиотечных подпрограмм общего назначения.

Математическое обеспечение графического дисплея СИГДА представляет собой библиотеку дисплейных подпрограмм на ЭВМ М-6000, которые можно разделить на четыре группы:

- административные подпрограммы;
- подпрограммы-генераторы графической информации;
- подпрограммы ввода информации с клавиатуры дисплея;
- управляющие подпрограммы.

Математическое обеспечение графического дисплея СИГДА на ЭВМ М-6000 описано в работах /3-5/.

Специальное математическое обеспечение предназначено для организации локального управления работой дисплейной станции, которая, с одной стороны, может использоваться как интеллектуальный графический терминал на линии с центральной ЭВМ, с другой - как автономная диалоговая вычислительная система, способная выполнять локальную обработку и графическое представление информации пользователя независимо от центральной ЭВМ.

Основными составными частями специального МО являются:

1. Блок иницирования.
2. Диспетчер УДС.
3. Интерпретатор запросов от БЭСМ-6.
4. Интерпретатор макрокоманд.
5. Интерпретатор таблицы действий.

Структура локального МО УДС показана на рис. 1.

### 2. Блок иницирования

Блок иницирования предназначен для начальной установки внутренних системных переменных, таблиц, очередей и стеков. Вызов этого блока происходит при начальном запуске УДС, а также в случае выдачи специального приказа для повторного иницирования УДС, который может быть передан со стороны центральной ЭВМ, либо введен пользователем с клавиатуры дисплея.

После завершения начальной установки блок иницирования вызывает диспетчер УДС, под управлением которого происходит вся дальнейшая работа дисплейной станции.

Блок иницирования реализован в виде отдельной подпрограммы длиной 50<sub>8</sub> слов.

### 3. Диспетчер УДС

Диспетчер УДС представляет собой управляющую программу, которая организует работу дисплейной станции, обеспечивая функционирование УДС в соответствии с командами, поступающими как от центральной ЭВМ, так и от человека.

Всякое действие, которое выполняется на УДС и носит функционально завершенный характер, будем в дальнейшем называть процессом. (Более подробно о понятии процесса см., например, /6/).

Все выполняемые на УДС процессы можно разделить на два типа.

К процессам первого типа относятся те, которые не связаны с выполнением операций ввода-вывода и не используют систему прерывания ЭВМ М-6000.

К процессам второго типа принадлежат процессы, связанные с выполнением операций ввода-вывода и использующие систему прерываний ЭВМ М-6000.

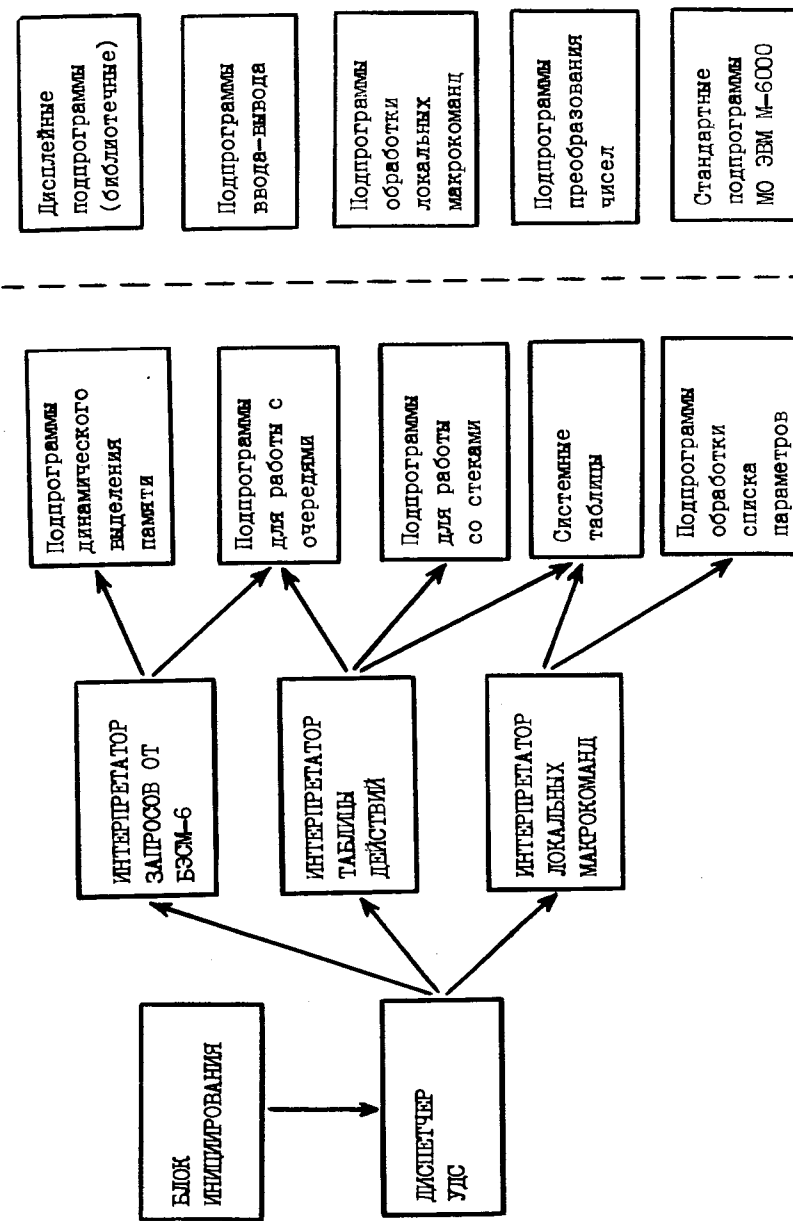


Рис. 1. Общая структура локального МО УДС на ЭВМ М-6000

Для организации обслуживания процессов диспетчер УДС использует две очереди.

Первая (входная) предназначена для приема информации, поступающей в УДС на обработку. Входная очередь имеет FIFO-структуру (т.е. дисциплину обслуживания: "первым вошел - первым вышел"). Предполагается, что все элементы входной очереди имеют одинаковый приоритет обслуживания.

Вторая очередь имеет LIFO-структуру (дисциплина обслуживания: "последним вошел - первым вышел"), и предназначена для запоминания состояния тех подпрограмм, выполнение которых временно прерывается.

Диспетчер удаленной дисплейной станции реализован в виде отдельной подпрограммы длиной 335<sub>8</sub> слов, блок-схема подпрограммы приведена на рис. 2.

### 3.1. Управление процессами

Диспетчер УДС управляет выполнением процесса, используя вектор его состояния. В общем случае вектор состояния содержит параметры процесса и специальный флаг, отражающий состояние выполняемого процесса во времени. Вектор состояния может быть различным в зависимости от типа процесса, при этом параметрами процесса могут быть адрес входа в исполнительную подпрограмму, обеспечивающую выполнение процесса (либо адрес продолжения в этой подпрограмме), требуемые ресурсы ввода-вывода, параметры обращения к подпрограмме и некоторые другие величины (например, промежуточные значения рабочих переменных подпрограммы). Параметры процесса фактически являются входными параметрами исполнительной подпрограммы, тогда как флаг состояния представляет собой выходной параметр, т.е. его значение устанавливается после выполнения подпрограммы. Кроме того, в качестве выходных параметров может быть задан вектор состояния другого процесса.

Выполнение любого процесса во времени разделяется на две фазы: фазу иницирования и фазу выполнения.

Считается, что процесс находится в фазе иницирования, если он характеризуется вектором состояния с ненулевым значением флага состояния процесса.

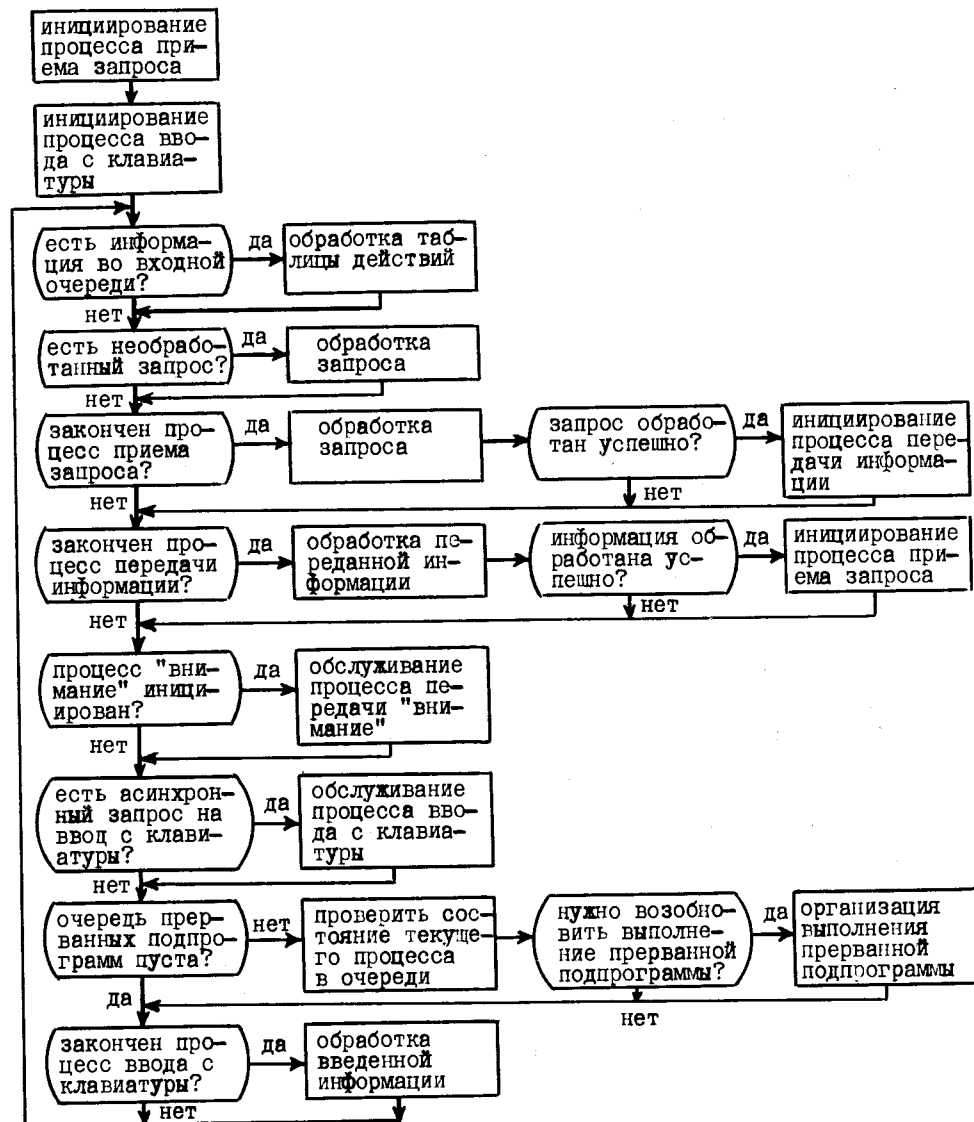


Рис. 2. Блок-схема диспетчера УДС

В фазе иницирования сам процесс непосредственно не выполняется, а фактически ожидает начала своего выполнения.

Диспетчер УДС циклически просматривает состояние всех управляемых им процессов и всякий раз, когда флаг состояния некоторого процесса принимает ненулевое значение, передает управление по адресу, указанному в векторе состояния. Подпрограмма, ответственная за выполнение процесса, проверяет возможность его реализации и, если требуемое действие не может быть выполнено в данный момент, то флаг состояния процесса сохраняет ненулевое значение, и управление возвращается диспетчеру УДС. В результате процесс остается в фазе иницирования до тех пор, пока не появятся условия для его выполнения. При этом диспетчер УДС будет в течение каждого цикла своей работы передавать управление в исполнительную подпрограмму. Таким образом, процесс может находиться в фазе иницирования, не препятствуя выполнению других процессов на УДС.

После появления условий для выполнения процесса флагу состояния присваивается нулевое значение, и выполнение процесса переходит во вторую фазу – фазу выполнения. В этой фазе происходит непосредственная реализация требуемого действия. Если процесс заканчивается ненормально, то флаг состояния принимает отрицательное значение, по которому диспетчер УДС организует обработку ненормально завершившегося процесса. В тех случаях, когда специфика ненормально закончившегося процесса позволяет осуществить его повторное выполнение, процесс иницируется заново. Это реализуется простой установкой флага состояния в единичное значение, остальные компоненты вектора состояния, как правило, не изменяются.

Основное различие между процессами первого и второго типов заключается в том, что в фазе выполнения процессов второго типа на время выполнения операции ввода-вывода происходит прерывание исполнительной подпрограммы, тогда как в случае процесса первого типа исполнительная подпрограмма не прерывается.

После того как при выполнении процесса второго типа иницируется операция ввода-вывода, флаг состояния устанавливается в нулевое значение, вектор текущего состояния процесса помещается в очередь прерванных подпрограмм и управление возвращается диспетчеру УДС.

Диспетчер УДС организует обслуживание очереди прерванных подпрограмм после того, как очередь становится непустой. При этом диспетчер УДС проверяет состояние обслуживаемого процесса на предмет окончания операции ввода-вывода, используя необходимую информацию, которая находится в векторе состояния процесса.

После появления условий, необходимых для возобновления выполнения исполнительной подпрограммы (в данном случае – после окончания операции ввода-вывода) диспетчер УДС удаляет вектор состояния текущего процесса из очереди и передает управление по адресу продолжения подпрограммы.

В общем случае в исполнительной подпрограмме процесса второго типа может присутствовать произвольное число операций ввода-вывода, при этом каждый раз на время выполнения ввода-вывода исполнительная подпрограмма прерывается.

Все операции ввода-вывода выполняются на ЭВМ М-6000 посредством системы прерываний независимо от работы диспетчера, и поэтому в фазе выполнения одновременно могут находиться несколько процессов второго типа.

Все процессы на УДС помимо их классификации по типам могут быть разделены на две группы:

1. Основные процессы.
2. Подчиненные процессы.

Основными являются те процессы, которые выполняют чисто системные функции, связанные с управлением УДС, и непосредственной обработки данных на ЭВМ М-6000 не осуществляют.

К основным относятся следующие процессы:

- прием запроса от центральной ЭВМ;
- обработка запроса от центральной ЭВМ;
- передача информации между ЭВМ, характер которой определяется запросом;
- обработка информации, переданной между ЭВМ;
- ввод информации с клавиатуры дисплея;
- обработка информации, введенной с клавиатуры дисплея;
- обработка входной очереди;
- обработка очереди прерванных подпрограмм;
- посылка сигнала "Внимание" в центральную ЭВМ.

Все системные процессы на УДС обслуживаются вне очереди и имеют равный приоритет обслуживания.

Подчиненные процессы появляются на УДС только в результате обработки входной очереди и их обслуживание происходит в соответствии с принятой дисциплиной обслуживания.

Подчиненные процессы образуют многочисленную группу различных процессов, непосредственно выполняющих функции локальной обработки данных. Если системные процессы в основном выполняются строго определенным образом и в известной последовательности, то характер и порядок выполнения подчиненных процессов заранее не известен, так как они зависят от пользователя. Удаленная дисплейная станция функционирует как интеллектуальный графический терминал, способный к нетривиальной обработке данных на ЭВМ М-6000 именно за счет выполнения подчиненных процессов на УДС.

В исходном состоянии диспетчер УДС инициирует выполнение следующих процессов второго типа:

- процесс приема запроса от центральной ЭВМ;
- процесс ввода информации с клавиатуры дисплея.

Выполнение этих процессов связано с внешними условиями и происходит по инициативе центральной ЭВМ и человека.

Окончание вышеуказанных процессов порождает новые процессы на УДС. Так, окончание процесса ввода с клавиатуры приводит к появлению процесса обработки введенной информации. Окончание процесса приема запроса приводит к появлению процесса его обработки, который в свою очередь порождает процесс обмена информацией между ЭВМ и т.д.

#### 4. Интерпретатор запросов от БЭСМ-6

Любая операция, связанная с передачей информации между центральной ЭВМ и УДС, разделяется на выполнение двух последовательных действий:

- центральная ЭВМ передает в УДС запрос на выполнение последующей операции обмена между ЭВМ;
- после окончания обработки запроса выполняется операция обмена между центральной ЭВМ и УДС.

Запрос содержит управляющую информацию, которая определяет характер последующей операции обмена и объем передаваемой инфор-

мации. Запрос передается в УДС отдельно из-за специфики работы канала связи между ЭВМ. В частности, передача информации между ЭВМ М-6000 и БЭСМ-6 происходит синхронным способом со скоростью передачи порядка 40 мкс/байт <sup>17</sup>/. При такой скорости работы канала процессор ЭВМ М-6000 не успевает выполнять одновременно с приемом информации также и ее обработку, поэтому эти операции выполняются раздельно.

Интерпретатор запросов реализован в виде подпрограммы длиной 520<sub>8</sub> слов, блок-схема его показана на рис. 4.

#### 4.1. Структура запроса

Запрос содержит фиксированное число слов и состоит из последовательности управляющих и информационных слов. Управляющее слово непосредственно определяет функцию запроса, информационные слова (если они определены) являются параметрами запроса. Структура управляющего слова запроса показана на рис. 3.

Код запроса занимает 4 разряда слова, допуская таким образом 16 возможных комбинаций. (В настоящее время используется только 9 типов запроса).

#### 4.2. Обслуживание запроса

Полное обслуживание запроса, переданного из центральной ЭВМ в УДС, реализуется посредством выполнения следующих четырех процессов:

- а) прием запроса;
- б) обработка принятого запроса;
- в) передача информации, определяемой характером запроса;
- г) обработка переданной информации.

В исходном состоянии после начального запуска УДС готова выполнить процесс приема запроса, передача которого происходит по инициативе центральной ЭВМ. После приема запроса подпрограмма-драйвер обмена устанавливает флаг процесса обработки запроса в ненулевое значение. Поэтому, когда процесс приема запроса заканчивается (при выходе из секции окончания драйвера обмена), то процесс обработки запроса оказывается уже инициализированным.

При обработке запроса возможны ситуации, когда выполнение

Код запроса	Функция запроса	Параметры запроса
0 0 0 0	Передача стандартного текстового сообщения	Длина текстового сообщения
0 0 0 1	Ввод текстовой информации с клавиатуры	_____
0 0 1 0	Передача таблицы действий в УДС	Длина таблицы действий
0 0 1 1	Чтение выходных параметров	_____
0 1 0 0	Обмен информацией между программами пользователя на УДС и БЭСМ-6	Начальный адрес Длина
0 1 0 1	Операция чтения-записи по адресу ЭВМ М-6000	Начальный адрес Длина
0 1 1 0	Передача графического объекта в УДС	Длина объекта Имя объекта Индикатор (акт./пасс.)
0 1 1 1	Чтение графического объекта из УДС	Имя объекта
1 0 0 0	Чтение выходной очереди "Внимание"	_____

Структура управляющего слова запроса

I	Не используются	Код запроса
---	-----------------	-------------

Рис. 3. Таблица запросов

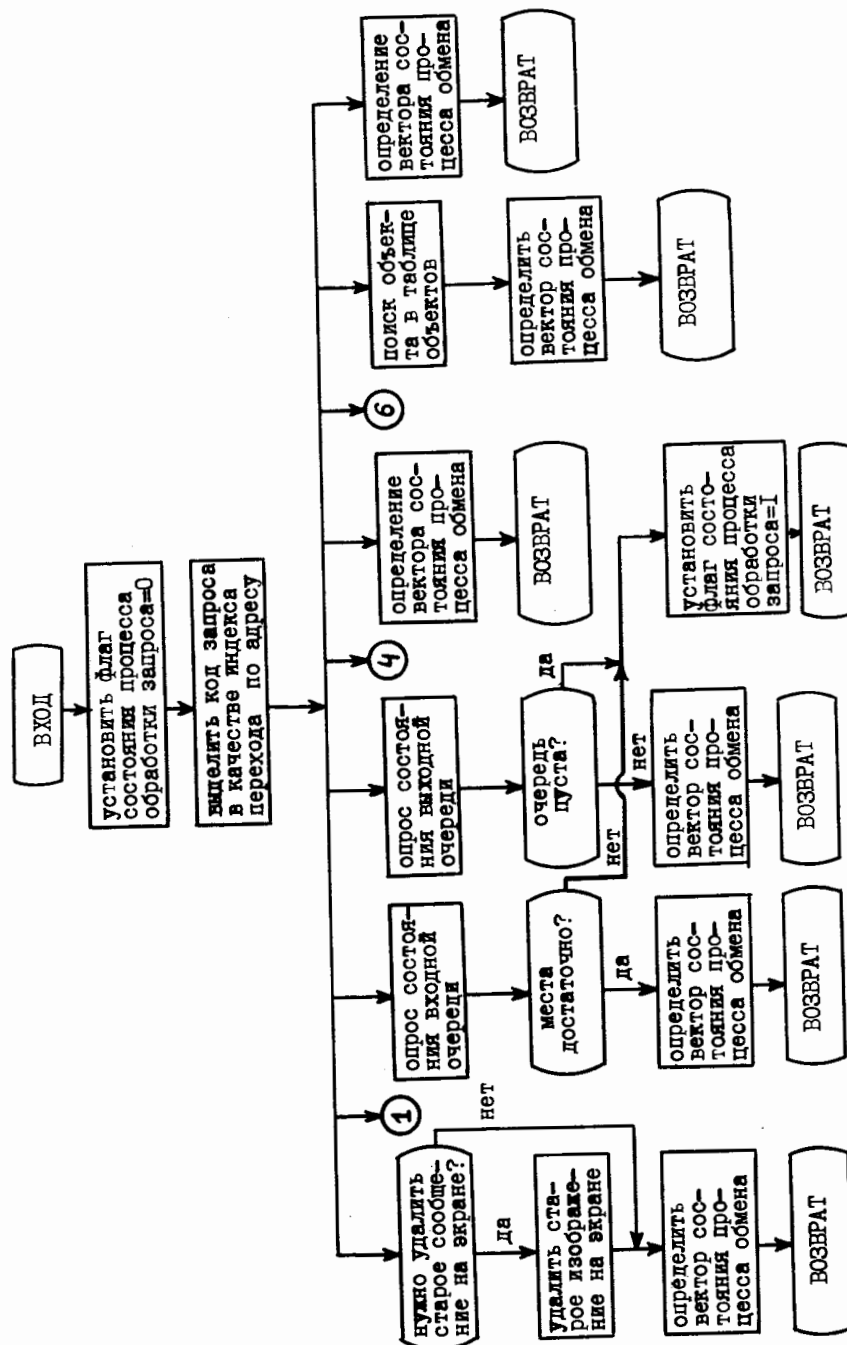


Рис. 4. Блок-схема интерпретатора запросов



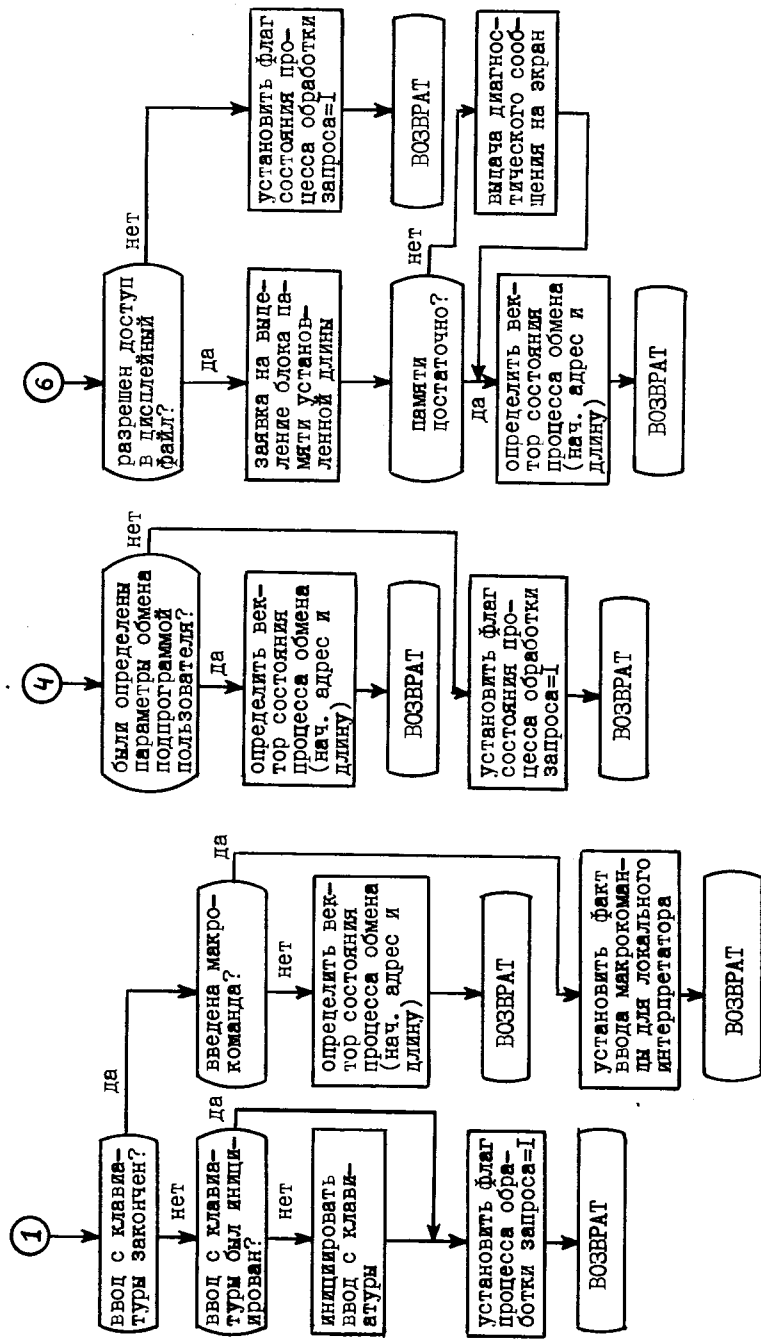


Рис. 4. Блок-схема интерпретатора запросов (продолжение)

требуемого действия не может быть организовано в данный момент из-за неготовности необходимого ресурса. Например, возможны следующие ситуации:

- центральная ЭВМ запрашивает ввод текстовой информации с клавиатуры, но ресурс занят (пользователь вводит информацию с клавиатуры);
- требуется прием таблицы действий, но во входной очереди нет достаточного места;
- центральная ЭВМ хочет передать в УДС некоторый объект и для его размещения в памяти ЭВМ М-6000 требуется выполнение процедуры "чистки мусора", которая временно запрещена до момента окончания операции ввода-вывода, выполняемой хотя бы с одним объектом.

Возможны также и другие случаи конкурирующих процессов при использовании одного и того же ресурса (см. рис. 4.). Во всех таких случаях обработка процесса задерживается (т.е. он в течение некоторого промежутка времени остается в фазе инициирования) до тех пор, пока требуемый ресурс не будет освобожден предшествующим процессом.

Входная информация для подпрограммы обработки запроса находится в буфере, выходными параметрами являются флаг состояния процесса обработки запроса, и если он обслужен успешно, то задается и вектор состояния для процесса последующего обмена информацией между ЭВМ. После окончания процесса обработки запроса иницируется процесс обмена информацией между ЭВМ. Исполнительная подпрограмма иницирует драйвер обмена, устанавливая флаг состояния в нулевое значение, и управление возвращается диспетчеру УДС. После этого процесс обмена информацией рассматривается диспетчером УДС как обслуженный (хотя в действительности он и не закончился). По окончании обмена, который происходит в режиме прерываний, флаг состояния процесса обработки переданной информации устанавливается в ненулевое значение подпрограммой-драйвером обмена (таким образом фиксируется факт окончания обмена) и на УДС иницируется процесс обработки переданной информации. Диспетчер УДС организует обслуживание процесса обработки переданной информации, вызывая соответствующую подпрограмму по ненулевому значению флага состояния этого процесса. Исполнительная подпрограмма реализует необходимое действие, которое зависит от

характера переданной информации, задает вектор состояния для процесса приема запроса, и процесс обработки переданной информации заканчивается. Далее диспетчер УДС организует обслуживание процесса приема запроса. По ненулевому значению флага состояния процесса происходит вызов подпрограммы-драйвера обмена и после этого УДС снова может принять запрос от центральной ЭВМ.

#### 5. Интерпретатор таблицы действий

Информация, предназначенная для локальной обработки, попадает во входную очередь в виде таблицы действий, которая может быть передана со стороны центральной ЭВМ, либо появиться во входной очереди как результат окончания процесса обработки макрокоманды, введенной с клавиатуры дисплея. Обработка входной очереди выполняется интерпретатором таблицы действий. Допустимая длина очереди определяется фактическим размером буфера в памяти ЭВМ М-6000 и длиной отдельных элементов, находящихся в очереди (элементом очереди является таблица действий, которая может быть переменной длины, обычно 5+20 16-разрядных слов).

Процесс обработки таблицы действий выполняется всякий раз в течение каждого цикла работы диспетчера УДС, когда входная очередь становится непустой. Интерпретирующая программа осуществляет обработку только одного элемента входной очереди, после чего управление возвращается диспетчеру УДС. Это необходимо для того, чтобы избежать возможных задержек при обслуживании диспетчером процессов на УДС в том случае, когда во входной очереди накапливается необработанная информация. После окончания процесса обработки таблица действий удаляется из входной очереди, и таким образом в очереди освобождается дополнительная память (в действительности никаких операций удаления данных физически не производится, изменяются лишь значения внутренних указателей очереди<sup>18/</sup>).

До окончания процесса обработки таблицы действий (который может быть процессом как первого, так и второго типа) обрабатываемая таблица действий остается во входной очереди и при необходимости процесс обработки может быть выполнен повторно.

#### 5.1. Структура таблицы действий

Таблица действий определяет некоторое действие, подлежащее

исполнению на УДС, и состоит из упорядоченной последовательности управляющих и информационных машинных слов.

Существует несколько типов управляющих слов:

1. Вызов подпрограммы.
2. Параметр.
3. Конец подпрограммы.
4. Конец таблицы действий.
5. Стандартное действие.
6. Режим ожидания.
7. Слово, определяемое пользователем.

Структура управляющих слов показана на рис. 5.

Управляющее слово "Вызов подпрограммы" определяет вызов подпрограммы на УДС. Последующие информационные слова указывают саму вызываемую подпрограмму, которая может быть задана значением адреса ее точки входа, либо именем подпрограммы в символьном виде.

Слово "Параметр" указывает на присутствие параметров обращения к вызываемой подпрограмме. Все последующие информационные слова являются параметрами обращения, которые могут быть заданы значениями, либо своими именами.

Управляющее слово "Конец подпрограммы" фиксирует последнее слово последовательности слов таблицы действий для вызова подпрограммы.

Слово "Конец таблицы действий" предназначено для окончания процесса обработки таблицы действий.

Слово "Стандартное действие" по функциональному назначению аналогично вызову подпрограммы, код стандартного действия непосредственно определяет вызываемую подпрограмму, которая всегда находится в памяти ЭВМ М-6000 независимо от версии локального МО УДС.

Слово "Режим ожидания" используется исключительно для системных целей и предназначено для временной блокировки операций обмена информацией между ЭВМ.

Управляющее слово, определяемое пользователем, служит для расширения функциональных возможностей УДС за счет предоставления пользователю системных средств для добавления собственных подпрограмм обработки таблицы действий в локальное МО УДС; функциональное назначение управляющего слова может быть различным по усмотрению самого пользователя.



Состояние очереди после завершения обработки макрокоманды

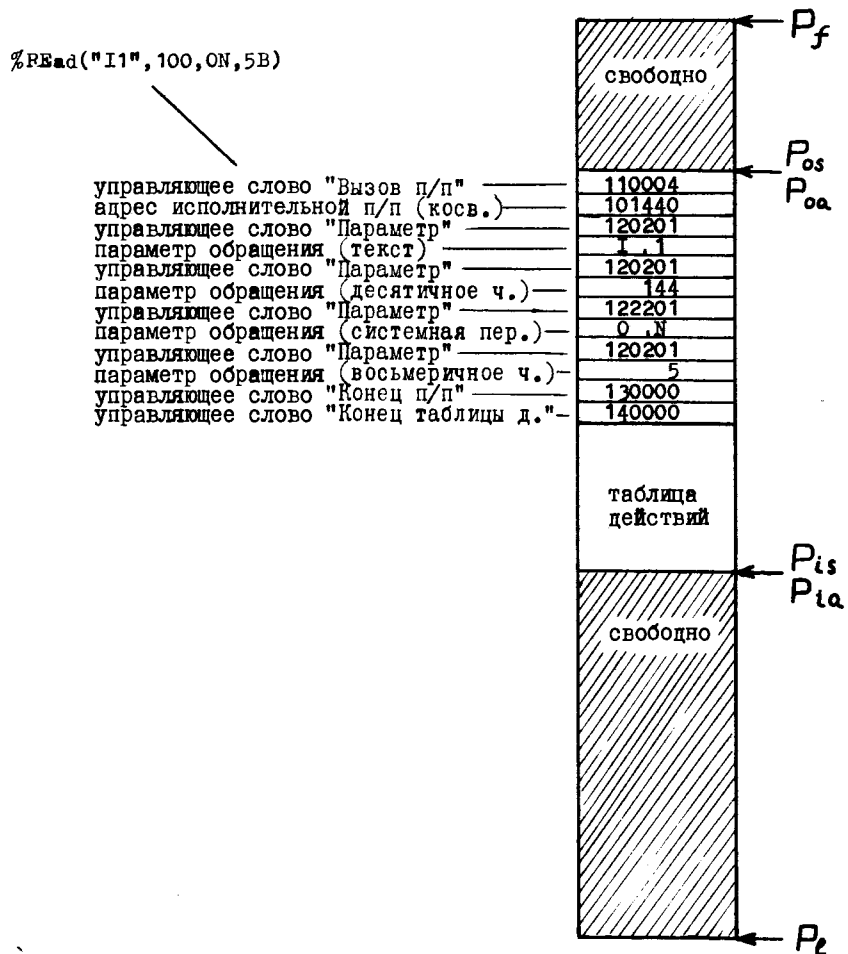


Рис. 6. Интерпретация макрокоманды и обработка таблицы действий в очереди

Состояние очереди в момент выполнения исполнительной подпрограммы

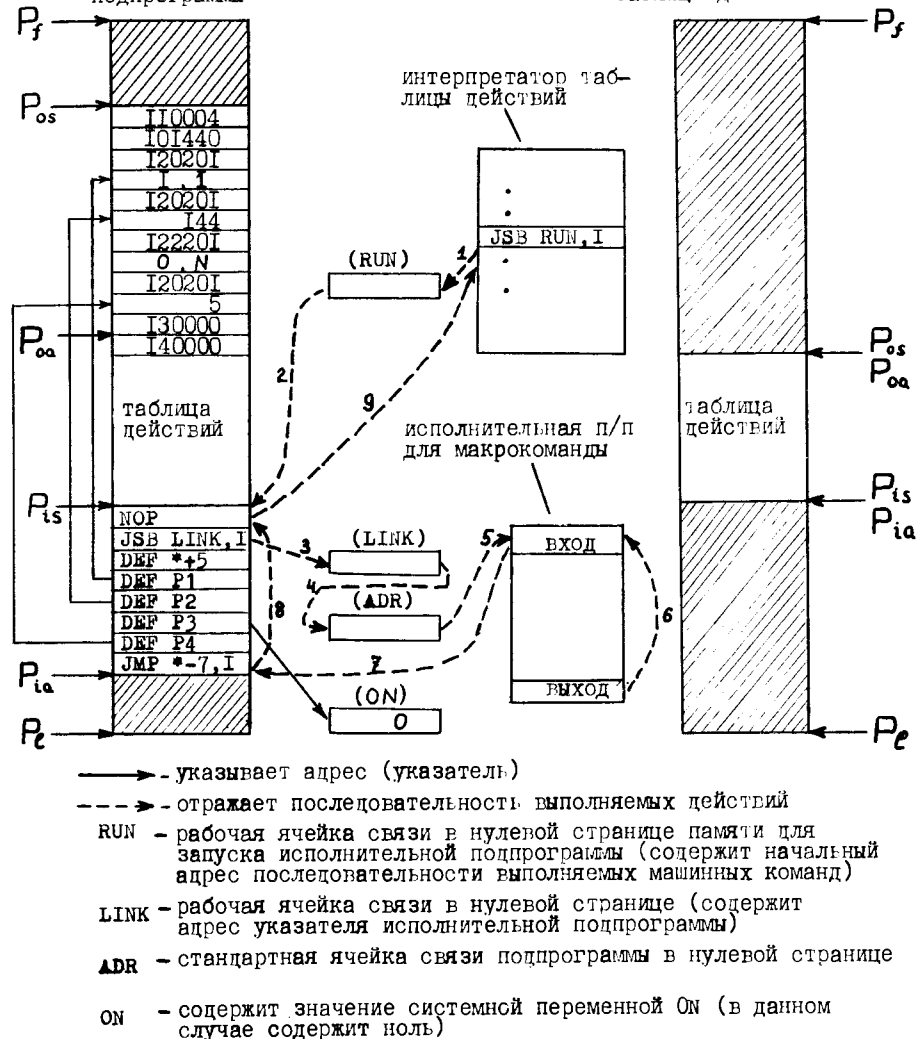


Рис. 6. Интерпретация макрокоманды и обработка таблицы действий в очереди (продолжение)

### 5.2.1. Выполнение подчиненного процесса первого типа

При вызове исполнительной подпрограммы возможны два случая: действие выполняется и действие не выполняется.

Если исполнительная подпрограмма не требует какого-либо ресурса (например, рабочий буфер из области динамически выделяемой памяти), то необходимое действие выполняется, флаг состояния процесса устанавливается в нулевое значение, управление возвращается интерпретатору, и процесс обработки таблицы действий продолжается.

В том случае, когда для реализации необходимого действия требуется некоторый ресурс, исполнительная подпрограмма проверяет возможность его использования. Если ресурс доступен, то действие выполняется. После окончания подчиненного процесса первого типа обработанная таблица действий и соответствующая ей вызывающая последовательность машинных команд удаляются из входной очереди.

Во втором случае необходимое действие не выполняется из-за занятости требуемого ресурса, и исполнительная подпрограмма возвращает управление интерпретатору с ненулевым значением флага состояния процесса. Интерпретатор удаляет из входной очереди последовательность машинных команд, соответствующих обрабатываемой таблице действий (которая остается в очереди), процесс обработки таблицы действий заканчивается, и управление возвращается диспетчеру УДС.

В течение последующих циклов работы диспетчера процесс обработки таблицы действий может повторяться многократно до тех пор, пока не освободится требуемый ресурс для выполнения подчиненного процесса.

### 5.2.2. Выполнение подчиненного процесса второго типа

Выполнение процесса происходит аналогичным образом, как и для процесса первого типа, но с некоторыми особенностями.

После инициирования операции ввода-вывода выполнение исполнительной подпрограммы прекращается, вектор состояния процесса помещается в очередь прерванных подпрограмм, флаг состояния обнуляется, управление возвращается интерпретатору, и процесс обработки таблицы действий продолжается. После окончания операции ввода-вывода, относящейся к данному процессу, диспетчер УДС во-

зобновляет выполнение прерванной подпрограммы. При этом управление передается по адресу продолжения (который запоминается в векторе состояния процесса), и выполнение прерванной подпрограммы продолжается.

Исполнительная подпрограмма, реализующая подчиненный процесс второго типа, может последовательно выполнять произвольное количество операций ввода-вывода. Каждый раз после инициирования операций ввода-вывода выполнение подпрограммы будет прерываться.

В случае заполнения очереди прерванных подпрограмм возможно появление дополнительных задержек в возобновлении выполнения прерванной подпрограммы. Дополнительная задержка возникает, начиная от момента окончания операции ввода-вывода, относящейся к прерванной подпрограмме, и до момента окончания выполнения прерванных подпрограмм, которые обслуживаются в очереди ранее.

Особенностью фазы окончания подчиненного процесса второго типа является то, что возврат из подпрограммы в диспетчер УДС происходит нестандартным способом.

Это объясняется тем, что стандартный возврат по адресу, определяемому положением вызывающей последовательности команд, недопустим из-за того, что состояние входной очереди к моменту возврата может измениться. (Например, команды вызывающей последовательности могут быть затерты новой таблицей действий).

## 6. Интерпретатор локальных макрокоманд

Клавиатура графического дисплея как устройство ввода алфавитно-цифровой и управляющей информации является основным инструментом диалога человека с ЭВМ на УДС. Процесс ввода информации с клавиатуры дисплея может быть инициирован как по запросу от центральной ЭВМ (синхронный ввод), так и по требованию оператора УДС (асинхронный ввод). После окончания процесса ввода с клавиатуры диспетчер УДС организует обработку введенной информации, которой может быть:

- локальная макрокоманда;
- информация для центральной ЭВМ.

Если первым введенным символом был знак процента (%), то информация интерпретируется как макрокоманда, в противоположном случае предполагается, что информация предназначена для центральной ЭВМ. Список локальных макрокоманд приведен в таблице I.

Таблица I.

Макрокоманда	Назначение
1	2
%Clear	Начальный запуск (иницирование УДС).
%Free	Уничтожение всех объектов пользователя (очистка дисплейного буфера).
%Delete(NAME)	Уничтожение объекта с именем NAME .
%Off(NAME)	Временное удаление (выключение) объекта с именем NAME.
%On(NAME)	Включение объекта с именем NAME .
%Pen	Вызов светового карандаша.
%Set(ADR,VAL)	Запись значения VAL по адресу ADR .
%Display(ADR)	Высвечивание на экране содержимого ADR динамически с возможностью просмотра содержимого памяти, используя функциональные клавиши.
%Attach(TABL)	Подключение таблицы переменных пользователя к системному МС.
%Dump	Вывод содержимого памяти УДС на бумажную ленту в абсолютном двоичном формате (системный дампинг).
%Output(NAME)	Вывод объекта NAME в символьном формате на алфавитно-цифровой дисплей.
%Output(NAME,UNIT)	Вывод объекта NAME в символьном формате на устройство UNIT .
%Punch(NAME)	Вывод объекта NAME в двоичном формате на ленточный перфоратор.
%Input(NAME,L)	Ввод символьной информации как объекта с именем NAME и предполагаемой длиной L с клавиатуры алфавитно-цифрового дисплея.
%Input(NAME,L,UNIT)	Ввод символьной информации как объекта с именем NAME и предполагаемой длиной L с устройства UNIT .
%Elektronix	Вывод графической картинке на запоминающий дисплей.

Продолжение

1	2
%Save	Запись графической картинке на магнитную ленту.
%Save(UNIT)	Вывод графической картинке на устройство UNIT .
%Load	Чтение графической картинке с магнитной ленты.
%Load(UNIT)	Чтение графической картинке с устройства UNIT .
%RWind	Перемотка ленты на начало.
%Skip	Пропуск одной логической записи вперед.
%Back	Пропуск одной логической записи назад.
%SFile	Пропуск одного файла вперед.
%BFile	Пропуск одного файла назад.
%EFile	Запись метки конца файла.
%Call(SUBR,P1,P2,...,PN)	Вызов подпрограммы SUBR с параметрами обращения P1, P2, ..., PN .
%Create(NAME,L)	Заявка на выделение памяти для активного графического объекта NAME с предполагаемой длиной L .
%Point(IX,IY)	Генерация невидимой точки с абсолютными координатами IX, IY .
%Dot(IDX,IDY)	Генерация видимой точки с относительными приращениями координат IDX, IDY .
%Vector(IDX,IDY)	Генерация вектора, заданного относительными приращениями координат IDX, IDY .
%Character('LINE OF TEXT')	Генерация символьной информации в положении текущих координат.
%Arc(R,A,B,M)	Генерация дуги радиусом R, началом A, конечным положением B, типа M (см. /4/).
%End	Конец генерации графического объекта, объявляемого посредством %CR .

## 6.1. Структура макрокоманды

Макрокоманда состоит из признака макрокоманды, идентификатора и, возможно, списка параметров. Идентификатор макрокоманды содержит не менее двух символов, список параметров заключается в скобки, параметры отделяются друг от друга через запятую.

Если первые два символа идентификатора введенной макрокоманды совпадают с одним из двухсимвольных идентификаторов в системной таблице /8/, то обработка макрокоманды выполняется на УДС.

Для задания параметров макрокоманды используются два способа: по значению и по имени переменной. В первом случае параметр может быть задан:

- целым числом (со знаком или без него);
- восьмеричным числом (со знаком или без него);
- текстовой константой;
- восьмеричным адресом.

Если используется способ задания параметра по имени, то в качестве параметра указывается символьный идентификатор системной переменной (который хранится в таблице /8/) или переменной пользователя.

Некоторые макрокоманды допускают задание переменного числа параметров, значения отсутствующих параметров принимаются по принципу умолчания.

## 6.2. Интерпретация локальной макрокоманды

После окончания ввода с клавиатуры подпрограмма-драйвер, осуществляющая ввод, устанавливает в ненулевое значение (равное количеству введенных символов) флаг состояния процесса обработки введенной информации и таким образом на УДС инициируется новый процесс.

Подпрограмма обработки информации, введенной с клавиатуры дисплея, передает управление интерпретатору локальных макрокоманд, если присутствует признак макрокоманды. Интерпретатор просматривает таблицу локальных макрокоманд и в случае успешного

поиска выполняет обработку введенной макрокоманды и генерирует таблицу действий во входной очереди.

В данном случае таблица действий определяет вызов подпрограммы, которая выполняет необходимую функцию макрокоманды. Если во время генерации таблицы действий происходит переполнение входной очереди, то процесс обработки макрокоманды прекращается, из входной очереди удаляется незавершенная таблица действий, флаг состояния процесса сохраняет ненулевое значение, управление возвращается диспетчеру УДС, и процесс обработки макрокоманды остается в фазе инициирования. После того как во входной очереди освободится достаточно места, процесс обработки локальной макрокоманды завершится нормально и будет сформирована соответствующая этой макрокоманде таблица действий.

Обработка этой таблицы действий выполняется интерпретатором таблицы действий как отдельный процесс, так, как это было описано ранее.

Пример интерпретации локальной макрокоманды показан на рис.6.

## 7. Управление особыми ситуациями

Диспетчер УДС обеспечивает параллельное обслуживание различных процессов, поэтому необходимо предусмотреть специальные меры для исключения критических ситуаций, связанных с выполнением процессов.

Возможны следующие особые ситуации:

1. Совместное появление нескольких процессов, которые требуют хотя бы один общий ресурс ввода-вывода.
2. Совместное появление нескольких процессов, каждый из которых имеет доступ во входную очередь и может изменить ее состояние.

В первом случае необходимо исключить возможность появления критической ситуации типа "dead lock". На УДС был реализован наиболее простой подход к решению этой проблемы. На все допустимые на УДС процессы накладывается ограничение в использовании не более одного ресурса ввода-вывода. При условии выполнения принятого ограничения критических ситуаций типа "dead lock" не возникает. Вместе с тем, принятое ограничение не является жестким, так как при необходимости выпол-

нения сложного действия, которое требует использования более чем одного ресурса ввода-вывода, реализация действия может быть разделена на выполнение последовательности отдельных процессов.

Во втором случае совместное появление нескольких процессов, каждый из которых может изменить состояние входной очереди, требует аккуратного программирования для критического интервала времени, в течение которого процесс имеет доступ во входную очередь.

Если некоторый процесс запрашивает доступ во входную очередь и получает его, то доступ во входную очередь закрывается на период критического интервала времени работы процесса.

### 8. Расширение функциональных возможностей УДС

Удаленная дисплейная станция обычно используется как интеллектуальный графический терминал типа "черный ящик" по отношению к задаче пользователя на стороне центральной ЭВМ /1,2/. В этом случае УДС располагает некоторым достаточно универсальным набором стандартных возможностей по локальной обработке данных на ЭВМ М-6000.

Вместе с тем локальное МО УДС позволяет расширить существующие функциональные возможности дисплейной станции за счет подпрограмм самого пользователя, которые он может подготовить для ЭВМ М-6000.

Предусмотрено несколько уровней доступа пользователя к локальному МО для расширения функциональных возможностей УДС.

1. Включение в состав локального МО дополнительных подпрограмм пользователя для реализации на УДС новых функций, посредством вызова этих подпрограмм или со стороны центральной ЭВМ, или автономно - с клавиатуры.

2. Расширение возможностей интерпретатора локальных макрокоманд за счет добавления новых подпрограмм.

3. Включение подпрограмм пользователя в локальное МО для обработки его собственной таблицы действий.

На первом уровне доступа пользователь загружает в ЭВМ М-6000 в дополнение к существующему локальному МО подпрограммы для реализации своих индивидуальных функций по локальной обработке данных на УДС. Чтобы получить доступ к подпрограммам пользователя по имени, требуется дополнительная подпрограмма-таблица, содержа-

щая идентификаторы переменных пользователя. Эта таблица имеет точно такую же структуру, как и таблица системных переменных /8/.

Таблица идентификаторов пользователя подключается к системному МО посредством вызова на ЭВМ М-6000 специальной подпрограммы (вызов реализуется макрокомандой %ATTACH).

При этом в качестве параметра обращения указывается начальный адрес таблицы идентификаторов пользователя (адрес определяется по таблице загрузки ЭВМ М-6000).

После подключения таблицы идентификаторов к локальному МО вызов необходимой подпрограммы пользователя на ЭВМ М-6000, как со стороны центральной ЭВМ (межпроцессорный вызов), так и автономно на УДС может быть выполнен по имени подпрограммы.

Для реализации межпроцессорного вызова в библиотеке подпрограмм МО УДС на БЭСМ-6 имеется специальная подпрограмма. Параметрами этой подпрограммы являются имя вызываемой на ЭВМ М-6000 подпрограммы в символьном виде и параметры обращения к ней.

Подпрограмма межпроцессорного вызова по заданным параметрам обращения генерирует таблицу действий и передает ее в УДС на обработку. Как результат обработки таблицы действий на УДС происходит вызов требуемой подпрограммы.

Автономный вызов подпрограммы пользователя на УДС может быть выполнен посредством соответствующей макрокоманды. Таким образом, на первом уровне доступа к локальному МО пользователю нужно знать правила написания подпрограммы-таблицы переменных пользователя и не требуется знания внутренней структуры МО УДС.

На втором уровне доступа пользователь пишет свои собственные подпрограммы для новых макрокоманд, добавляемых к локальному МО.

В этом случае пользователю необходимо изменить одну системную подпрограмму на УДС: таблицу локальных макрокоманд. Для этого в таблицу добавляются элементы, описывающие новые макрокоманды.

При желании пользователь может удалить из таблицы те макрокоманды, которые он считает ненужными для его области применения УДС. На втором уровне доступа пользователю достаточно знать структуру таблицы макрокоманд /8/.

Третий уровень доступа дает возможность пользователю определять его собственную таблицу действий. В частности, пользова-



тель определяет новые управляющие слова таблицы действий и пишет подпрограммы для реализации функций, определяемых назначением управляющих слов. Третий уровень доступа в отличие от первого дает средство для расширения функциональных возможностей УДС за счет добавления новых возможностей локального МО ЭВМ М-6000.

На третьем уровне доступа к локальному МО УДС необходимо знать внутреннюю структуру данных УДС и процедуры обработки и управления данными, находящихся в очереди.

#### 9. Заключение

За период опытной эксплуатации удаленной дисплейной станции с 1975 года локальное МО прошло несколько циклов перевоплощений, при этом каждый раз расширялись возможности УДС для локальной обработки данных. Эффективное управление УДС вместе с наличием гибких структур данных в локальном МО позволяют легко расширять и изменять возможности дисплейной станции в зависимости от характера применений и от оснащения ЭВМ М-6000 периферийным оборудованием.

#### Литература

1. Кавченко А.В. и др. ОИЯИ, РЮ-9325, Дубна, 1975.
2. Заикин Н.С. и др. ОИЯИ, РЮ-9326, Дубна, 1975.
3. Кавченко А.В. и др. УСИМ, 1974, № 1, с. 110.
4. Кавченко А.В. и др. ОИЯИ, Б1-11-7654, Дубна, 1974.
5. Кавченко А.В. и др. Организация данных для графического дисплея. - В кн.: Совещание по программированию и математическим методам решения физических задач. ОИЯИ, ДЮ-7707, Дубна, 1974, с. 332.
6. Колин А. Введение в операционные системы. "Мир", М., 1975.

7. Заикин Н.С. Математическое обеспечение линий связи БЭСМ-6 с периферийными ЭВМ. - В кн.: Совещание по программированию и математическим методам решения физических задач. ОИЯИ, Дубна, ДЮ-7707, Дубна, 1974, с. 346.
8. Карлов А.А., Полынецев А.Д. ОИЯИ, ДЮ-10946 Дубна, 1977.

Рукопись поступила в издательский отдел  
6 октября 1977 года.