

Ц 8419
К-238

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

ДУБНА

2/1-78



122/2 - 78

10 - 10946

А.А.Карлов, А.Д.Полынецв

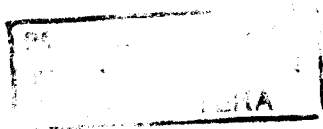
СТРУКТУРЫ ДАННЫХ
УДАЛЕННОЙ ДИСПЛЕЙНОЙ СТАНЦИИ

1977

10 - 10946

А.А.Карлов, А.Д.Полынецв

СТРУКТУРЫ ДАННЫХ
УДАЛЕННОЙ ДИСПЛЕЙНОЙ СТАНЦИИ



Карлов А.А., Полиншев А.Д.

10 - 10946

Структуры данных удаленной дисплейной станции

В работе дается описание структур данных, используемых в локальном математическом обеспечении удаленной дисплейной станции на базе малой ЭВМ-6000, подключенной к центральной ЭВМ БЭСМ-6 через канал связи.

Приводится описание цепочек, очередей, стеков, таблиц и даются способы хранения информации и методы доступа к ней. Все описываемые в работе средства реализованы в виде набора отдельных подпрограмм, которые могут представлять интерес для пользователей ЭВМ М-6000.

Работы выполнены в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1977

I. Введение

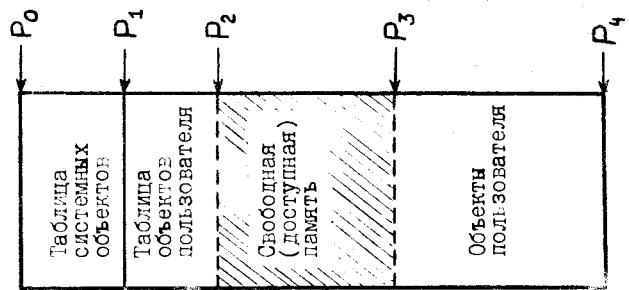
При разработке локального математического обеспечения удаленной дисплейной станции (МО УДС) /1/ на базе ЭВМ М-6000 вопросы выбора структуры данных имеют существенно важное значение и в конечном счете определяют гибкость и эффективность использования малой ЭВМ.

Дисплейная станция, выполняя различные функции по приему, накоплению и обработке информации, поступающей как от центральной ЭВМ, так и от человека, выступает по отношению к задаче пользователя как "черный ящик" /2-3/.

Характер и объем поступающей в УДС информации, вообще говоря, заранее не определен. Поэтому основными требованиями, предъявляемыми к внутренней организации данных УДС, являются возможность динамического выделения памяти и возможность работы с очередями.

Задача динамического выделения памяти в условиях диалоговой работы на УДС обусловлена необходимостью работы с графическими объектами (когда в процессе диалога наряду с появлением на УДС новых объектов требуется уничтожение старых) и организации на УДС приема информации как от центральной ЭВМ, так и от пользователя дисплейной станции, для чего требуется временное выделение памяти, объем которой заранее не известен.

Возможность работы с очередями позволяет организовать асинхронный режим работы УДС как по отношению к центральной ЭВМ, так и по отношению к пользователю. В этом случае центральная ЭВМ может передавать в УДС очередную порцию информации, не ожидая окончания обработки переданной ранее информации, в результате не требуется жесткой синхронизации выполнения программ на двух независимых процессорах (центральной ЭВМ и УДС). Со своей стороны, пользователь может вводить новую информацию, не ожидая окончания обработки той информации, которая была введена им ранее.



Структура элемента таблицы объектов (ЭТО)

M	Адрес объекта
	Длина объекта
	Имя объекта

M - маска объекта (0 - активный, 1 - пассивный)

Рис. 1. Структура области динамического выделения памяти.

Предусмотрены два типа очередей: очереди, имеющие FIFO - структуру (с дисциплиной обслуживания "первым вошел - первым вышел"), и очереди, имеющие LIFO - структуру (с дисциплиной обслуживания "последним вошел - первым вышел").

Предлагаемые структуры данных позволяют исключительно эффективно и гибко выполнять операции одновременного накопления и обработки данных, не заботясь о способах хранения информации в памяти ЭВМ и о методах доступа к ней.

2. Динамическое выделение памяти

В условиях ограниченных ресурсов оперативной памяти малой ЭВМ М-6000 для УДС был предложен достаточно простой алгоритм динамического выделения памяти. Реализация этого алгоритма требует относительно небольших системных издержек на программирование. Программы динамического выделения памяти в основном используются для работы с графическими объектами, однако допускается возможность работы с объектами любой природы при условии, что они идентифицируются посредством имен аналогично графическим объектам ^{14/}.

Свободной памятью объявляется неиспользуемая область оперативной памяти (буфер), которая остается после загрузки всех подпрограмм локального МО ЭВМ М-6000.

Буфер состоит из трех частей, размер которых изменяется динамически, а границы определяются специальными указателями (рис.1). Верхнюю часть буфера занимает таблица объектов ^{14/}, сами объекты находятся в нижней части буфера, а средняя его часть является свободной (доступной) для использования.

Отдельный элемент таблицы объектов состоит из трех слов и содержит начальный адрес, длину и имя объекта. Знаковый разряд первого слова элемента указывает, является ли данный объект активным или замаскированным. Все объекты неграфической природы регистрируются в таблице объектов как замаскированные.

Таблица объектов, в свою очередь, состоит из двух частей. Первая имеет фиксированную длину (в зависимости от версии локального МО) и предназначена для таблицы системных графических объектов (диагностические и пояснительные сообщения и т.п.). Эта таблица формируется при начальном запуске системы. Вторая часть

таблицы имеет переменную длину и представляет собой таблицу объектов пользователя.

Состояние буфера описывается следующими указателями:

P_0 - начальный адрес области динамического выделения памяти (является также начальным адресом таблицы системных объектов);

P_1 - начальный адрес таблицы объектов пользователя;

P_2 - адрес первого слова доступной для использования памяти;

P_3 - адрес последнего слова доступной для использования памяти;

P_4 - адрес последнего слова области динамического выделения памяти.

Указатели P_2 и P_3 являются динамическими, P_0 , P_1 и P_4 - фиксированными. Значение указателей P_0 и P_4 определяется после окончания загрузки подпрограммы локального МО в ЭВМ М-6000, а значение указателя P_1 - после формирования таблицы системных объектов.

Аппарат динамического выделения памяти предоставляет пользователю следующие возможности:

- выделение блока памяти заданной длины;
- отказ от использования выделенного ранее блока памяти;
- определение текущего положения выделенного для использования блока памяти.

В заявке на выделение памяти указывается необходимая длина блока памяти, а также имя, под которым выделяемая память будет зарегистрирована в таблице объектов. При обслуживании такой заявки проверяется возможность выделения памяти из свободной области, ограниченной указателями P_2 и P_3 . Если свободной памяти достаточно, то определяется начальный адрес для размещения нового объекта, корректируется указатель P_3 и в таблицу объектов добавляется новый элемент, который содержит параметры объекта (начальный адрес, длину, имя). Если элемент был добавлен в таблицу объектов за счет ее расширения (в таблице не было пустых мест), то корректируется также и указатель P_2 . Если смежной области памяти, ограниченной указателями P_2 и P_3 оказалось недостаточно, то выполняется процедура "чистки мусора". Все объекты, зарегистрированные в таблице

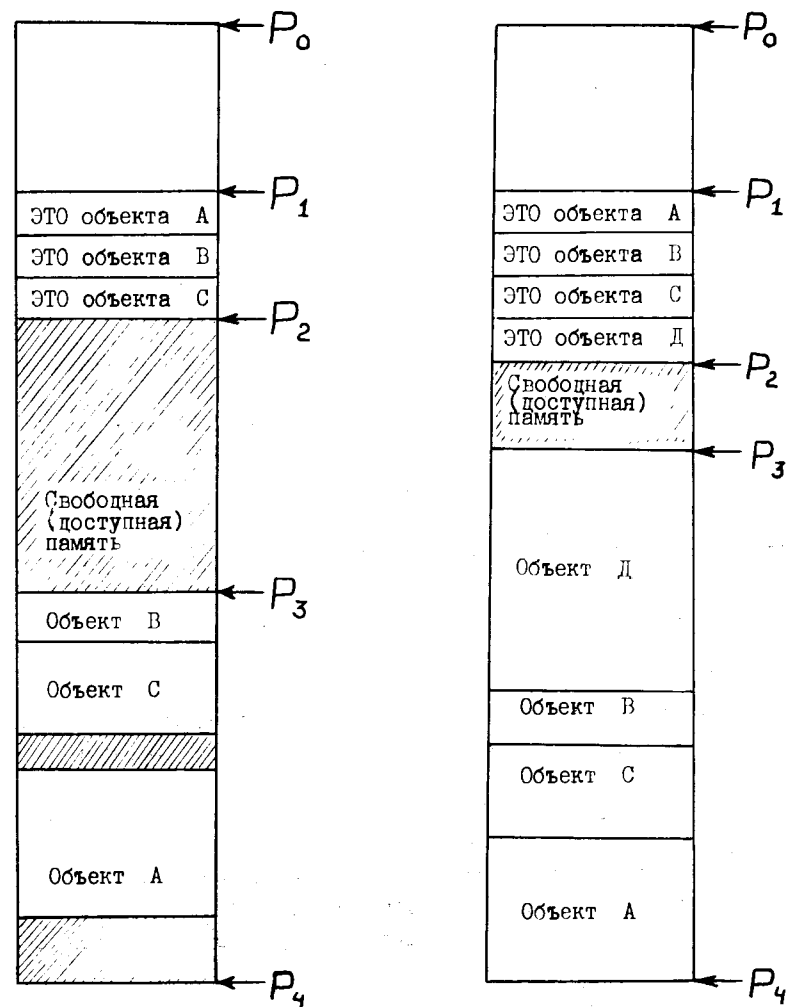


Рис. 2. Пример динамического выделения памяти для объекта Д с выполнением операции "чистка мусора".

объектов пользователя, по возможности (т.е. при наличии памяти, освобожденной за счет удаления отдельных объектов) перемещаются в нижнюю часть буфера так, чтобы между объектами не оставалось пустых мест. При этом в таблице объектов корректируются адреса сдвинутых объектов. Аналогичным образом в верхней части буфера перемещаются элементы таблицы объектов пользователя.

После уплотнения буфера корректируются указатели P_2 и P_3 и доступная для пользования область памяти представляет собой блок смежной памяти максимально возможного размера. Если и после этого доступной памяти оказывается недостаточно, то в буфере выделяется область максимально возможной длины.

В случае отказа от использования некоторого блока выделенной ранее памяти (как объекта с заданным именем) в таблице объектов уничтожается соответствующий элемент; никаких действий, связанных с появлением "мусора" в буфере, при этом не выполняется.

При опросе текущего положения выделенного блока памяти определяется начальный адрес соответствующего объекта, который выбирается из таблицы объектов пользователя.

Пример динамического выделения памяти с выполнением операции "чистки мусора" показан на рис. 2.

3. FIFO - структуры

На ЭВМ М-6000 был реализован набор подпрограмм для работы с динамическими структурами, имеющими дисциплину обслуживания: "первым вошел - первым вышел" (FIFO - структуры, или нормальные очереди). Принятая для УДС организация FIFO - структур аналогична описанной в работе [5].

Для организации FIFO - структуры выделяется некоторый буфер, который состоит из заголовка и рабочей области (рис. 3). Заголовок содержит следующие внутренние указатели и индикаторы для организации FIFO - структуры:

- P_f - указатель первого слова очереди;
- P_e - указатель последнего слова очереди;
- P_{ia} - "активный" указатель ввода; всегда указывает позицию слова в очереди для помещения очередного элемента в очередь;

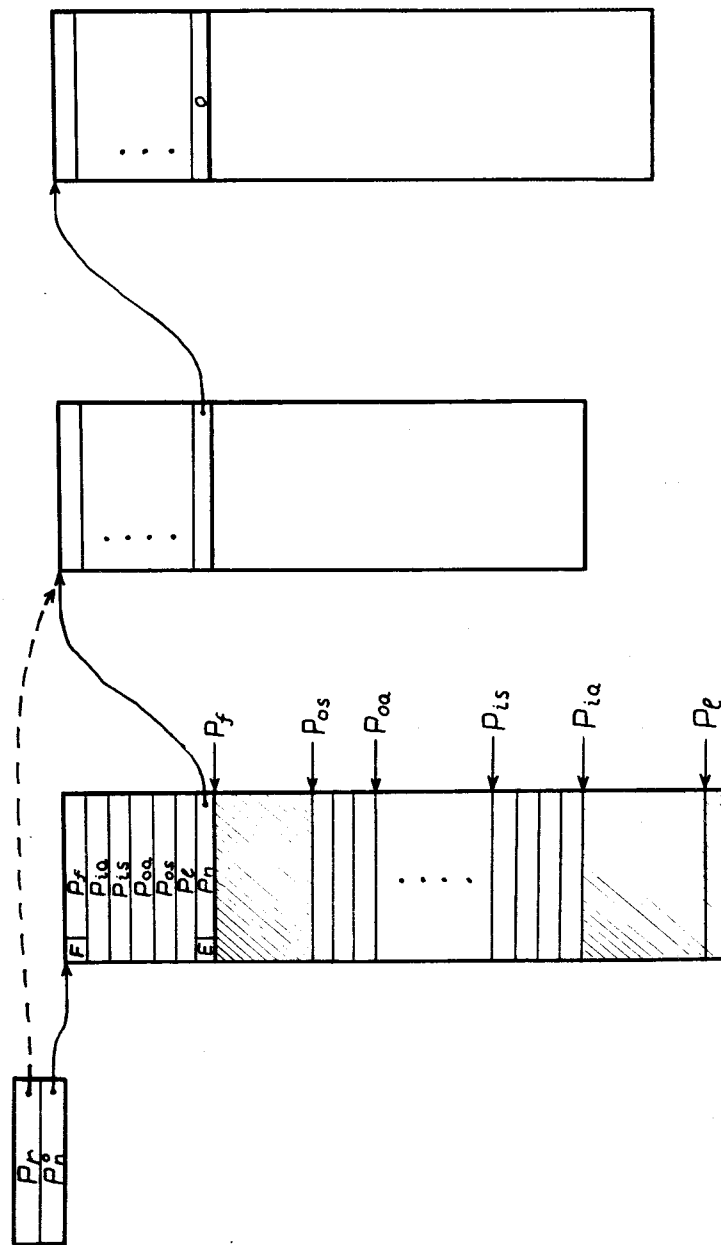


Рис. 3. Целочисленные структуры данных и FIFO-структуры МО УДС.

P_{is} - "статический" указатель ввода; остается неизменным до тех пор, пока не будет закончена текущая операция ввода данных в очередь; указывает текущую начальную позицию данных, помещаемых в очередь;

P_{oa} - "активный" указатель вывода; всегда указывает позицию слова, извлекаемого из очереди;

P_{os} - "статический" указатель вывода; остается неизменным до тех пор, пока не будет закончена текущая операция вывода данных из очереди на обработку; указывает текущую начальную позицию данных, извлекаемых из очереди;

F - индикатор заполненной очереди (если $F = 1$, то очередь заполнена);

E - индикатор пустой очереди (если $E = 1$, то очередь пуста);

Возможны три состояния очереди:

1) очередь пуста ($F = 0, E = 1$);

2) очередь заполнена ($F = 1, E = 0$);

3) очередь частично заполнена ($F = 0, E = 0$).

Комбинация $F = 1$ и $E = 1$ не допускается, как противоречивая.

Наличие двух индикаторов позволяет определить состояние очереди, не выполняя над ней никаких операций (в работе /5/ предлагалось использовать только один индикатор, в результате чего состояние очереди можно было определить только динамически при выполнении операции с очередью).

Кроме внутренних указателей и индикаторов, в заголовке находится внешний указатель P_n для связи очередей в цепочку (см. ниже).

Для работы с очередью был реализован ряд процедур;

1. Объяснить указанный буфер FIFO - структурой. Формируется заголовок очереди, все внутренние указатели устанавливаются в исходное состояние.

2. Поместить 16-разрядное слово в очередь. Слово помещается в позицию, указанную P_{ia} , указатель P_{ia} продвигается на единицу, либо выполняется циркуляция адреса ($P_{ia} = P_f$). Указатель P_{ia} может двигаться только до указателя P_{os} .

3. Достать слово из очереди. Извлекается слово из позиции, указанной P_{oa} , указатель P_{oa} увеличивается на единицу, либо выполняется циркуляция адреса ($P_{oa} = P_f$); указатель P_{oa} может двигаться только до указателя P_{is} .

4. Зафиксировать "статический" указатель ввода P_{is} . Указателю P_{is} присваивается значение P_{ia} . Таким образом, информация, которая находилась ранее между указателями P_{ia} и P_{is} , становится доступной для операции 3.

5. Восстановить "активный" указатель ввода P_{ia} . Указатель P_{ia} принимает значение P_{is} . Информация, которая ранее находилась в очереди между указателями P_{ia} и P_{is} , теряется.

6. Зафиксировать "статический" указатель вывода P_{os} . Указатель P_{os} принимает значение P_{oa} . В результате в очереди освобождается место, ограниченное ранее указателями P_{oa} и P_{os} .

7. Восстановить "активный" указатель вывода P_{oa} . Указатель P_{oa} принимает значение P_{os} . Информация, которая ранее находилась между указателями P_{oa} и P_{os} , становится вновь доступной для операции 3.

8. Возвратить "активный" указатель ввода P_{ia} на одну позицию. Указатель P_{ia} уменьшается на единицу, либо происходит циркуляция адреса, и P_{ia} присваивается значение P_e . Указатель P_{ia} может двигаться только до указателя P_{is} .

9. Возвратить "активный" указатель вывода P_{oa} на одну позицию. Указатель P_{oa} уменьшается на единицу, либо происходит циркуляция адреса, и P_{oa} присваивается значение P_e . Указатель P_{oa} может двигаться только до указателя P_{os} .

На УДС имеется три отдельных очереди:

1. Входная очередь таблиц действий, передаваемых от центральной ЭВМ, а также генерируемых на УДС интерпретатором локальных макрокоманд.

2. Выходная очередь параметров, возвращаемых в центральную ЭВМ после обработки таблиц действий.

3. Выходная очередь символьной информации, введенной пользователем с клавиатуры и предназначенной для центральной ЭВМ.

4. Цепочки

Перечисленные выше очереди, имеющие FIFO - структуру, связаны между собой посредством внешних указателей и образуют цепочную структуру данных /6/.

Для организации цепочной структуры данных имеются следующие указатели:

P_n^0 - указатель первого блока цепочки;

P_n^i - указатель, связывающий i -й блок цепочки с $(i+1)$ -м ($i = 1, 2, \dots, N-1$, где N - число блоков в цепочке), при этом $P_n^N = 0$;

P_z - указатель блока цепочки, выбранного для текущей обработки; для i -го блока $P_z = P_n^{i-1}$.

Структура цепочки показана на рис. 3.

Для работы с цепочной структурой данных предусмотрены следующие процедуры:

1. Включение отдельного блока, объявленного ранее очередью, в цепочку. Блок добавляется в конец цепочки, устанавливаются значения двух указателей: P_n^i для предшествующего блока и нулевое значение указателя P_n^{i+1} для последнего блока цепочки.

2. Переход по цепочке вперед (от i -го блока к $(i+1)$ -му). Устанавливается новое значение указателя P_z , равное P_n^i .

3. Переход по цепочке назад (от i -го блока к $(i-1)$ -му). Устанавливается новое значение указателя P_z , равное P_n^{i-2} .

Для выполнения операции переходов по цепочке был использован алгоритм поиска, основанный на "переворачивании" указателей при прохождении по цепочке вперед с последующим восстановлением значений указателей при обратном прохождении /7/.

Все процедуры для работы с FIFO - структурами и цепочками реализованы в виде набора подпрограмм общей длиной 550₈ слов и могут найти применение в различных областях программирования на ЭВМ М-6000.

5. LIFO - СТРУКТУРЫ

Разновидностью очереди являются динамические структуры, имеющие дисциплину обслуживания: "последним вошел - первым вышел" (LIFO - структуры, или стеки).

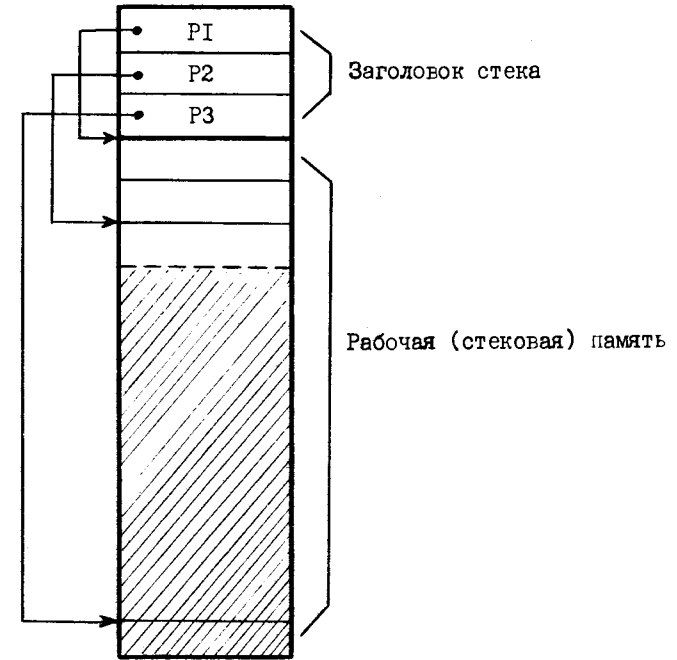


Рис. 4. Структура стека (пример: в стек помещено три слова).

Все стеки, используемые в локальном МО УДС, имеют структуру, которая показана на рис. 4.

Чтобы допустить возможность одновременной работы с несколькими стеками, все указатели стековой памяти располагаются в заголовке самого стека. Указатели стека имеют следующий смысл:

- P_1 - указатель первого слова стековой памяти;
- P_2 - указатель текущей позиции стека (указывает на последнее слово, помещенное в стек);
- P_3 - указатель последнего слова стековой памяти.

Помимо указателей, при работе со стеками используется специальный индикатор M , указывающий динамическое состояние стека. Назначение индикатора M будет показано при описании процедур.

Для работы со стеками предусмотрены следующие процедуры:

1. Объявить указанный буфер стеком. Формируется заголовок стека, указатели стека устанавливаются в исходное состояние.

2. Поместить слово в текущий стек. Если $P_2 \neq P_3$, т.е. если стек не заполнен, то указатель P_2 продвигается на единицу, слово помещается в позицию, указанную P_2 , и устанавливается значение индикатора $M = 0$. Если $P_2 = P_3$, т.е. когда стек заполнен, то слово в стек не заносится и устанавливается значение индикатора $M = 1$.

3. Извлечь слово из текущего стека. Если $P_2 \neq P_1 - 1$, т.е. если стек не пуст, то извлекается слово из позиции, указанной P_2 , указатель P_2 уменьшается на единицу и устанавливается значение индикатора $M = 0$. В случае, если $P_2 = P_1 - 1$, т.е. стек пуст, устанавливается значение индикатора $M = 1$.

4. Объявить текущим стеком один из существующих стеков. Все последующие операции типа 2 и 3 будут выполняться в указанном стеке до момента переопределения текущего стека посредством процедур 1 и 4. Указатели стека сохраняют свои прежние значения.

5. Очистить указанный стек. Указатель P_2 данного стека принимает значение $P_1 - 1$. Никаких других действий не выполняется.

Индикатор динамического состояния стека (M) задается как параметр обращения к подпрограммам, реализующим операции типа 2 и 3, и доступен для последующего анализа после выполнения этих операций.

Таблица системных переменных (пример)

D	.	S
D	.	E
L	.	
1	1	0 2 3
D	.	S
0	.	N
1	1	0 7 1
.	.	.
.	.	.
.	.	.

5-символьный идентификатор переменной (DSDEL)
 адрес переменной
 4-словный элемент таблицы

Таблица локальных макрокоманд (пример)

D	.	I
0	6	0 3 4
1	0	0 0 0 1
S	.	I
0	0	0 0 0 1
0	0	0 0 0 0
0	.	U
1	0	0 0 1 2
1	5	3 6 1
.	.	.
.	.	.

2-символьный идентификатор макрокоманды
 адрес п/п, либо код стандартного действия
 слово-спецификация для подпрограммы
 3-словный элемент таблицы

Слово-спецификация

M	Не используется	T	P
---	-----------------	---	---

- M - индикатор способа вызова подпрограммы;
 $M = 0$, п/п определяется кодом стандартного действия;
 $M = 1$, п/п определяется адресом точки входа.
- T - тип исполняемой подпрограммы;
 $T = 0$, п/п имеет фиксированное число параметров обращения;
 $T = 1$, п/п имеет переменное число параметров.
- P - число параметров обращения к подпрограмме.

В локальном МО УДС стеки используются для организации очереди прерванных подпрограмм /I/, а также при выполнении реентерабельных подпрограмм.

Все процедуры, предусмотренные для работы со стеками, реализованы в виде набора подпрограмм общей длиной 152₈ слова. Эти подпрограммы включены в расширенную библиотеку стандартных подпрограмм ЭВМ М-6000 и доступны для других применений.

6. Системные таблицы

В локальном МО УДС имеется две системных таблицы: таблица системных переменных и таблица локальных макрокоманд.

Первая таблица содержит имена системных переменных в символьном виде и адреса этих переменных. Каждый элемент таблицы состоит из четырех слов: первые три слова содержат пятисимвольный идентификатор переменной, четвертое слово содержит значение адреса самой переменной. Структура таблицы показана на стр.15.

В качестве системных переменных могут выступать как имена отдельных подпрограмм локального МО, так и имена внутренних переменных различных подпрограмм. Таблица системных переменных содержит всю необходимую информацию для организации доступа к переменным по именам. В частности, это позволяет при выполнении межпроцессорного обращения со стороны центральной ЭВМ к подпрограмме, которая находится в памяти ЭВМ М-6000, указывать имя вызываемой подпрограммы, не заботясь о ее физическом расположении в памяти.

В том случае, если пользователь расширяет функциональные возможности УДС за счет написания собственных подпрограмм обработки данных на ЭВМ М-6000 /I/, то он может определить и таблицу собственных переменных. Таблица переменных пользователя может содержать имена его собственных подпрограмм или отдельных переменных. Например, если пользователь добавляет в локальное МО подпрограммы с именами USER 1 , USER 2 , ..., USER N , то таблица переменных пользователя на языке мнемокода ЭВМ М-6000 может быть задана в виде, представленном на стр.17.

Таблица переменных пользователя включается в системное МО УДС посредством специальной локальной макрокоманды (% ATTACH /I/).

Таблица переменных пользователя (пример)

```
ASMB,R,L,T,B
NAM USTAB
```

```
?
? (комментарии для подпрограммы)
? эта подпрограмма представляет собой пример таблицы
? переменных пользователя; в данном случае таблица
? содержит информацию о подпрограммах пользователя.
?
```

```
EXT USER1,USER2,...,USERN (описывает внешние п/п)
ASC 3,USER1 (символьное имя п/п)
DEF USER1 (адрес точки входа в п/п)
ASC 3,USER2
DEF USER2
.
.
.
ASC 3,USERN
DEF USERN
DEC -1 (признак конца таблицы)
END
```

Таблица локальных макрокоманд содержит информацию, необходимую для интерпретатора локальных макрокоманд. Структура этой таблицы показана на стр. 15.

Каждый элемент таблицы состоит из трех слов и содержит символичный идентификатор макрокоманды (переме два символа), адрес подпрограммы для выполнения макрокоманды, а также управляющее слово, которое определяет тип и структуру макрокоманды.

Расширение набора локальных макрокоманд на УДС выполняется за счет соответствующей модификации таблицы локальных макрокоманд посредством добавления в нее новых элементов.

7. Заключение

Структуры данных локального МО УДС являются универсальными системными средствами и могут найти применение как при решении отдельных задач пользователями, так и при разработке математического обеспечения системными программистами.

Л И Т Е Р А Т У Р А

1. Карлов А.А., Польшингер А.Д. ОИЯИ, II-10967, Дубна, 1977.
2. Катченко А.В. и др. ОИЯИ, P10-9325, Дубна, 1975.
3. Заикин Н.С. и др. ОИЯИ, P10-9326, Дубна, 1975.
4. Катченко А.В. и др. Организация данных для графического дисплея. В кн.: Совещание по программированию и математическим методам решения физических задач, ОИЯИ, D10-7707, Дубна, 1974, с.332.
5. Christofferson J.L. Datamation, 1973, No 19, P.87.
6. Польшингер А.Д. Структуры данных. В кн.: Международная школа по вопросам использования ЭВМ в ядерных исследованиях. ОИЯИ, D10, II-8450, Дубна, 1974, с.217.
7. Schorr H., Waite W.M. Communication of the ACM, 1969, 10, P.501.

Рукопись поступила в издательский отдел
5 сентября 1977 года.