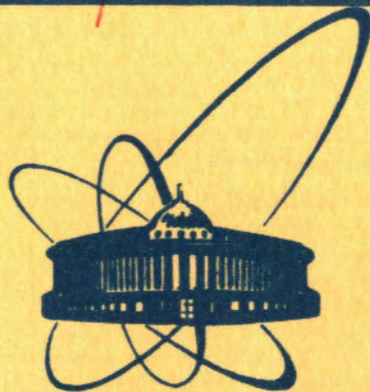


2840/2-80

23/н-80



сообщения ⁺
объединенного
института
ядерных
исследований
дубна

P11-80-202

А.А.Карлов, А.С.Квирлов,

STRING - ДИАЛОГОВАЯ ПОДСИСТЕМА ДЛЯ
СОГЛАСОВАННОГО ВЫВОДА ТЕКСТОВЫХ
ДАННЫХ НА ГРАФИЧЕСКИЙ ТЕРМИНАЛ

1980

1. ВВЕДЕНИЕ

Основным преимуществом графических диалоговых систем перед неграфиче-

скими является возможность вести диалог с пользователем при помощи "картинок": чертежей, графиков, рисунков, что обеспечивает, в частности, более высокую скорость восприятия информации человеком по сравнению с выводом печатного текста. Однако очевидно, что и "текстовая" форма представления данных при графическом диалоге, безусловно, необходима. Объектами, требующими "текстового" представления /в дальнейшем такие объекты будут именоваться просто текстами/ являются таблицы, диагностические сообщения, справочная информация и т.п.

В общем случае задача вывода и размещения текстов на экране графического дисплея является более сложной, чем та же задача для рисунков. Рисунок, как правило, занимает весь или большую часть экрана. В этом случае для размещения следующего рисунка достаточно предварительно стереть предыдущий. Для текстов ситуация иная. Их размеры меняются от одной строки до сотен и более, поэтому на экране в одном случае могут поместиться несколько текстов, а в другом - может не хватить места и для одного.

На начальном этапе развития математического обеспечения удаленной дисплейной станции /УДС/ ОИЯИ^{1/} вся работа по организации диалога, в том числе по выводу текстов и согласованию их взаимного расположения на экране, возлагалась на пользователя. Он обладал полной информацией о создаваемой им программе и имел возможность предупредить все конфликтные ситуации, о чем и должен был заботиться. По мере развития математического обеспечения УДС пользователь все больше освобождается от забот, не связанных непосредственно с программированием основного алгоритма его задачи^{2/}. С появлением специализированных диалоговых подсистем /напр. ^{3/} - автономных программных средств, доступ к которым пользователь может осуществлять с терминала в произвольный момент диалога без необходимости предварительного программирования этого доступа в своей программе, он освободился от программирования стандартных операций по представлению и анализу изображений в диалоговом режиме. В результате он утратил возможность управлять согласованным выводом текстов, так как по отношению к пользователю отдельная подсистема является "черным ящиком" с фиксированным набором функциональных возможностей. С другой стороны, разработчик подсистем также не может обеспе-

чить согласованный в рамках всей задачи вывод текстов, так как моменты и порядок вызова подсистем и собственных подпрограмм выбираются пользователем по его усмотрению и не могут быть предусмотрены системным программистом.

Поэтому возникает необходимость в разработке методики и программных средств для согласованного вывода текстов, которые были бы общими для всех подсистем и подпрограмм как служебных, так и разработанных пользователем.

Эта методика должна обеспечить компактный вывод текстов с оптимальным заполнением экрана, защиту текстов от взаимных наложений и случайных стираний. Она должна также обеспечить возможность просмотра больших текстов, размер которых превышает емкость экрана, через малое "окно" / в том числе, если оно составляет только часть экрана/.

2. ОБЩАЯ МЕТОДИКА

Естественной структурой, упорядочивающей и согласующей размещение текстов, является страница, которая может занимать весь экран или его часть. Страница характеризуется числом строк, длиной строки и величиной интервала между строками /интервал между символами определяется техническими параметрами дисплея/. Расположение страницы на экране задается координатами начала первой строки.

Страница является своего рода "общим буфером" для всех источников текстов. Тексты располагаются на странице сверху вниз в порядке вывода. Каждый текст выводится с новой свободной строки, чем обеспечивается защита текстов от взаимных наложений. Совокупность строк страницы, занятых одним текстом, назовем абзацем. Длина абзаца определяется числом составляющих его строк и может меняться от 1 строки до размера страницы.

В процессе вывода текстов страница постепенно заполняется, и может наступить момент ее "переполнения" /т.е. возникает ситуация, когда после заполнения последней строки страницы вывод текста оказывается еще не законченным/. Для обработки этого случая однозначное решение отсутствует, а поэтому решение задачи нельзя полностью поручить системе. Действительно, прежде всего нужно освободить место для продолжения вывода. Стирать всю страницу можно лишь с разрешения пользователя, чтобы предотвратить утерю ценной информации. Особое положение занимает последний абзац, так как вывод соответствующего текста еще не закончен и, в принципе, может быть даже повторен. Выборочно стирать другие абзацы неразумно, это может запутать пользователя. Наконец, можно просто прекратить вывод прерванного текста, ничего не стирая.

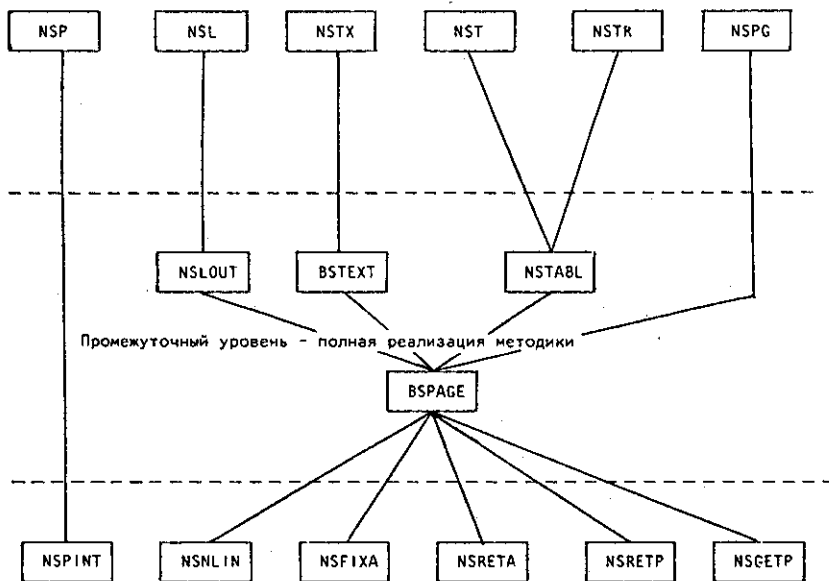
В этой ситуации для выбора правильного решения требуется участие пользователя, что и предусмотрено в данной реализации. В случае переполнения страницы дальнейший вывод текста производится частями /блоками/, размер которых совпадает с размером последнего абзаца. Предусмотрены процедуры последовательной /в прямом и обратном направлении/ и выборочной выдачи частей текста.

3. РЕАЛИЗАЦИЯ ОБЩЕЙ МЕТОДИКИ

Для реализации описанной выше общей методики создана диалоговая подсистема STRING (Subsystem of Text

Representation for Interactive Graphics). Ее подпрограммы образуют иерархическую структуру, состоящую из трех уровней/см. рис.1/.

Верхний уровень - диалоговый интерфейс



Нижний уровень - операции над страницей

Рис.1. Иерархия подпрограмм диалоговой подсистемы STRING.

Нижний уровень образуют подпрограммы, предназначенные для выполнения отдельных операций над страницей /объявление страницы, ее очистка и т.п./. Пользователь подсистемы в общем случае не обязан знать о существовании подпрограмм этого уровня, за исключением, возможно, подпрограммы объявления страницы (NSPINT) и подпрограммы-функции, обеспечивающей возможность доступа к значениям параметров страницы (NSGETP). Изучение подпрограмм нижнего уровня может потребоваться лишь в том случае, если у пользователя возникает необходимость запрограммировать процесс вывода текстов самостоятельно, например, для исключения в целях экономии памяти тех или иных возможностей общей методики.

Перечень и краткое описание подпрограмм нижнего уровня даны в приложении.

Промежуточный уровень образуют подпрограммы, реализующие общую методику вывода текстов в полном объеме. Эти подпрограммы доступны пользователю через общепринятый механизм вызова библиотечных подпрограмм /оператор CALL на ФОРТРАНе/. Помимо практической стороны, создание подпрограмм промежуточного уровня целесообразно также и с методической точки зрения, для того, чтобы отделить средства реализации общей методики от средств обеспечения диалогового доступа к ним. Более подробно подпрограммы промежуточного уровня описаны в п.п. 6 и 7.

Верхний уровень подсистемы образуют подпрограммы, предназначенные для организации диалогового доступа пользователю непосредственно с терминала к подпрограммам промежуточного уровня с помощью системного приказа /С / аналога оператора CALL/^{4/}. Наличие этого уровня имеет принципиальное значение, поскольку именно оно и превращает совокупность подпрограмм, реализующих описываемую методику, в диалоговую подсистему.

Описание подпрограмм верхнего уровня приведено в п.8.

4. ОБЪЯВЛЕНИЕ СТРАНИЦЫ

При работе в рамках диалогового монитора^{/2/} объявление страницы производится при его инициировании, и параметры страницы получают следующие значения: число строк - 25, интервал - 30 ед. раstra, длина строки - 64 символа, координаты начала первой строки - /0,840/.

Указанные значения являются стандартными и действительны по умолчанию. Пользователь может объявить страницу с другими параметрами, если стандартные значения его не удовлетворяют. Порядок объявления страницы приведен в приложении. При прекращении задач пользователя и возвращении на системный уровень диалоговый монитор восстанавливает стандартные значения параметров страницы.

При работе без диалогового монитора, т.е. когда верхний уровень не используется, объявление страницы в программе пользователя является обязательным.

5. ОБРАБОТКА ПЕРЕПОЛНЕНИЯ СТРАНИЦЫ Обнаружив попытку "переполнить" страницу, система обращается к пользователю, который должен оценить ситуацию и с помощью одной из команд /см. табл.1/ сообщить системе свое решение.

Таблица 1

Команда	Действие
C	- продолжить выдачу текста
B	- вывести предыдущую часть текста
n	- вывести n-ную часть текста, где n = 1,2,..
L	- вывести последнюю часть текста
CL	- очистить страницу
RTN	- прекратить процедуру вывода текста

Все команды, кроме CL и RTN, предварительно очищают последний абзац, а затем используют его для продолжения вывода. При этом ранее выведенные на экран тексты сохраняются без изменений. Если же их содержимое не представляет ценности, пользователь может очистить страницу с помощью команды CL, а затем продолжить вывод текущего текста на всю страницу.

Таким образом, предложенный набор команд реализует операции последовательного и выборочного доступа к тексту. Он удобен для просмотра большого текста через малое "окно" /абзац/.

6. ВЫВОД ТЕКСТОВ Рассмотрим три наиболее типичных случая вывода текстов.

Вывод одной строки

Для вывода текста в виде одной строки предусмотрена программа
NSLOUT (TEXT, NCH),

где TEXT - массив с текстовой информацией;
NCH - число выводимых символов.

Следует помнить, что максимально допустимая длина строки на УДС составляет 64 символа. Это определяется техническими возможностями графического дисплея СИГДА.

Вывод текста как массива символов

Для случая, когда необходимо осуществить вывод текста, представляющего собой массив символов объемом более одной строки, предусмотрена подпрограмма:

BSTEXT (ТЕХТ, LT),

где **ТЕХТ** - массив с текстовой информацией;

LT - его длина.

При работе этой подпрограммы очередная строка символов для представления на экране выбирается исходя из заданной длины строки в странице. В случае работы с этой подпрограммой пользователь должен заранее предусмотреть в массиве наличие соответствующего числа пробелов, если необходимо обеспечить "выравнивание" строк или какие-либо другие особенности при выводе текста.

Вывод текста по формату пользователя

Если у пользователя возникает необходимость самому формировать информацию для каждой строки текста по его собственному алгоритму, то он может воспользоваться подпрограммой:

BSPAGE (NSUB),

где **NSUB** - текстовая константа, указывающая имя вспомогательной подпрограммы, через которую подпрограмма **BSPAGE** осуществляет доступ к тексту.

Подпрограмма **NSUB** составляется пользователем и имеет три параметра:

NL - номер очередной строки;

IB - массив, в который следует поместить очередную выбранную строку;

NC - длина строки.

Массив **IB** описан в подпрограмме **BSPAGE** как массив длиной 11. Для выдачи очередной строки на экран подпрограмма обращается к подпрограмме с именем **NSUB**, которая должна поместить в массив **IB** строку текста с номером **NL** и указать ее длину через параметр **NC**. В случае некорректности параметра **NL** /выход за пределы текста/ подпрограмма **NSUB** должна возвращать отрицательное значение параметра **NC**.

При составлении подпрограммы NSUB пользователю могут потребоваться значения некоторых параметров страницы. Доступ к этим параметрам он может получить через подпрограмму-функцию NSGETP /см. приложение/.

Следует подчеркнуть, что наличие подпрограммы BSPAGE открывает для пользователя возможность гибкой компоновки текста, включая в него, например, строки с результатами декодировки текущих значений переменных. В общем случае каждая отдельная строка единого текста может быть сформирована по своим собственным правилам.

В работе подпрограмм NSLOUT, BSTEXT и BSPAGE имеется общая особенность. Если текст, выводимый с помощью одной из этих подпрограмм, полностью поместился в свободной части страницы, то выход из подпрограммы производится автоматически после окончания вывода текста. В противном случае в момент переполнения страницы будет инициирован диалог /см. п.5/, и выход из подпрограммы будет произведен только после ввода пользователем команды RTN.

Объем памяти, необходимый для работы подпрограмм NSLOUT, BSTEXT, BSPAGE и используемых ими, составляет около 3К. Кроме того, для размещения графической информации необходим текущий буфер^{5/}, длина которого может быть вычислена по формуле:

$$((NL * LL) / 2 + 4 * NL) / 3 + 1,$$

где NL - число строк в странице;

LL - число символов в строке.

7. ТАБЛИЧНОЕ ПРЕДСТАВЛЕНИЕ ОДНОМЕРНЫХ МАССИВОВ

Табличное представление числовых массивов - одна из распространенных задач, где может успешно применяться предлагаемая общая методика вывода текстов.

Для табличного представления одномерных массивов была создана подпрограмма NSTABL, которая обеспечивает преобразование числовых /целых, вещественных, восьмеричных/ данных в алфавитно-цифровую форму и вывод полученного текста по правилам общей методики. При этом, если формат преобразования не указан пользователем при обращении к подпрограмме, он выбирается автоматически. Подпрограмма NSTABL может также быть использована для вывода массивов, содержащих текстовые данные.

Обращение к подпрограмме производится следующим образом:

CALL NSTABL (A, L, ITP, NM, KFM, KFD, KDG),

- где A - идентификатор массива данных;
 L - длина массива;
 ITP - текстовая константа, задающая тип данных / IHA - вещественный, IHM - целый, IHT - текстовый, прочее - восьмеричный/;
 NM - "имя" массива - произвольная текстовая константа, высвечиваемая на экране для идентификации массива;
 KFM - спецификация формата преобразования /например, 4HF6.1 /; если KFM = 0, необходимая спецификация будет выбрана автоматически;
 KFD - длина поля, задаваемого спецификацией; если KFM = 0, значение KFD не играет роли;
 KDG - количество значащих цифр в представлении числа, используемое при автоматическом выборе формата для вещественного массива.

Общее количество памяти, необходимое для работы подпрограммы NSTABL и ею используемых, составляет около 3,5 К.

Для работы подпрограммы необходим текущий буфер, размер которого вычисляется по формуле, приведенной в п.6.

8. ДИАЛоговый ИНТЕРФЕЙС

Как было указано выше, подпрограммы верхнего уровня предназначены для организации доступа пользователю к подпрограммам промежуточного уровня в режиме диалога /см. рис.1/. Эти подпрограммы в подсистеме STRING играют роль своего рода программного интерфейса между пользователем и самой подсистемой. Наличие подпрограмм верхнего уровня позволяет освободить пользователя от необходимости программирования процедур вывода текстов на дисплей в своей программе.

Для запуска с терминала подпрограмм верхнего уровня, как указывалось, используется системный приказ /C, доступный в любой момент диалога. Общий формат этого приказа^{4/}:

/C. < имя подпрограммы >, < список параметров > .

В качестве параметров используются константы и имена диалоговых переменных^{6/}.

Доступные с терминала процедуры, имена соответствующих подпрограмм верхнего уровня и параметры этих подпрограмм приведены в табл.2.

Таблица 2

№	Назначение	Имя подпрограммы	Параметры
1.	Объявление страницы	NSP	1 - координата X начала первой строки, 2 - координата Y начала первой строки, 3 - интервал между строками, 4 - число строк в странице, 5 - число символов в строке
2.	Вывод строки символов	NSL	1 - строка символов, 2 - число символов в строке,
3.	Вывод текста	NSTX	1 - имя диалогового массива, содержащего текст
4.	Вывод текста по формату пользователя	NSPG	1 - имя подпрограммы пользователя / заданное в виде текстовой константы/, которая формирует требуемый формат
5.	Построение таблицы значений одномерного массива	NST	1 - имя диалогового массива
6.	Построение таблицы значений одномерного массива вещественных чисел	NSTR	1 - имя диалогового массива, 2 - число значащих цифр в представлении элементов массива

Пример диалога с использованием подсистемы STRING приведен в следующем разделе.

9. ПРИМЕР

Предположим, что необходимо проанализировать некоторый числовой массив YFUNCT длины 258. Для анализа требуется представить на экране дисплея и график, и таблицу значений массива, причем необходимо просматривать эту таблицу, имея график постоянно на

экране. Поэтому график и таблица /или часть ее/ должны присутствовать на экране одновременно. Предположим также, что перед таблицей значений массива на экран должно быть выведено некоторое пояснительное сообщение /текст/. Для определенности график должен быть расположен в верхней части экрана, а пояснительное сообщение и таблица должны занимать оставшуюся его часть.

Выполнить указанные требования можно несколькими способами, например, так, как это сделано в ходе диалога, протокол которого приведен на рис.2.

Прежде всего в верхней части экрана с помощью подсистемы SING^{/3/} построен искомый график. Затем в свободной от графика области экрана с помощью п/п NSP объявлена страница, на которую далее с помощью п/п NSTX выведено пояснительное сообще-

/A,SING	}	Построение графика YFUNCT(X) с помощью диалоговой подсистемы SING.
SCR,200,900,750,900		
AXL,'F6.1','F6.1'		
GRF,YFUNCT,X		
/R		
/C,NSP,0,720,24,25,64		Объявление страницы
/C,NSTX,TEXT		Вывод сообщения
/C,NSTR,YFUNCT,3		Вывод таблицы
C		Продолжение вывода таблицы
CL		Очистка страницы
C		Продолжение вывода таблицы
RTN		Прекращение вывода таблицы

Примечание: YFUNCT,X,TEXT - имена диалоговых массивов, объявленных в программе пользователя.

Рис.2. Протокол диалога.

щение, хранящееся в диалоговом массиве TEXT. Так как сообщение целиком поместилось на странице, выход из п/п NSTX произведен автоматически.

Вслед за этим для построения таблицы значений массива была вызвана п/п NSTR. В процессе вывода таблицы эта подпрограм-

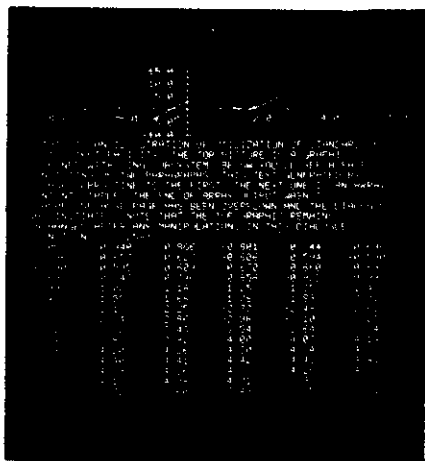


Рис. 3

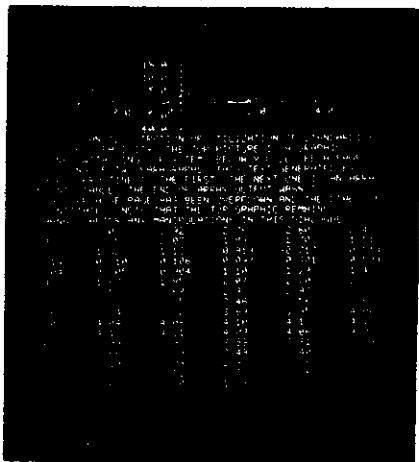


Рис. 4

ма обнаружила попытку переполнения страницы и инициировала диалог, описанный в п.5. На рис.3 приведено изображение графика, текста и таблицы, которое было на экране в этот момент. Для продолжения выдачи таблицы введена команда C. Результат этого действия приведен на рис.4. Как видно из рисунка, и график, и текст /составляющий первый абзац страницы/ остались без изменений. Чтобы удалить пояснительное сообщение и использовать всю страницу для выдачи только таблицы, введена команда CL, а затем вновь команда C /см. результат на рис.5/. Для окончания диалога /выхода из п/п NSTR /введена команда RTN.

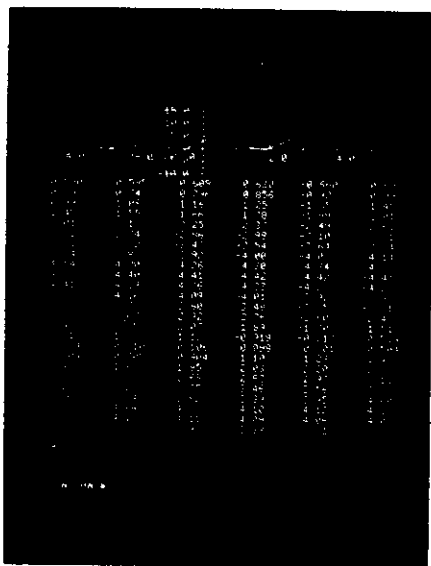


Рис.5

10. ЗАКЛЮЧЕНИЕ

Предлагаемая общая методика предоставляет гибкие и универсальные средства, предназначенные для организации согласованного вывода текстов на графический терминал системными и пользовательскими подпрограммами независимо от порядка их вызова. В результате обеспечивается: защита текстов от взаимных наложений и разрушения, компактное заполнение заданной части экрана /страницы/ текстовой информацией, а также организация просмотра больших массивов через малое "окно" в диалоговом режиме.

Методика реализована в виде диалоговой подсистемы подпрограмм, которая позволяет организовать вывод текстов в диалоговом режиме без программирования этого вывода в прикладной программе. Трехуровневая иерархическая структура подсистемы допускает самостоятельное применение отдельных операций методики путем использования подпрограмм соответствующего уровня.

ПРИЛОЖЕНИЕ

Ниже приведено описание подпрограмм нижнего уровня, реализующих набор элементарных операций над страницей. Если пользователь по каким-либо причинам сочтет нецелесообразным применение подпрограмм более высоких уровней, он может запрограммировать выдачу текста самостоятельно с помощью подпрограмм из данного набора. Чтобы выполнить при этом требования общей методики, он должен придерживаться следующих правил:

1. Выдаче каждого текста должно предшествовать объявление нового абзаца /п/п NSFIXA /.
2. Тексты выводятся построчно. Для получения номера и координат очередной строки следует использовать подпрограмму NSNLIN. Она же используется для определения момента переполнения страницы.
3. При переполнении страницы механизм выдачи координат и номеров строк блокируется. Для его разблокировки и продолжения вывода текста нужно освободить или последний абзац /п/п NSRETA/ или всю страницу /п/п NSRETP /.
4. Все действия, связанные с генерацией графической информации, представляющей текст, а также по управлению соответствующими графическими объектами, возлагаются на пользователя.
5. Текст, соответствующий каждому абзацу, является отдельным графическим объектом. Номер этого объекта определяется по формуле: $3000 + K$, где K - номер абзаца.

Имеются следующие подпрограммы нижнего уровня:

1. NSPINT - объявление страницы.

Обращение:

CALL NSPINT (IX, IY, LSTEP, NLIN, NCHAR, ER),

где IX , IY - координаты начала первой строки, LSTEP - интервал между строками, NLIN - число строк в странице, NCHAR - длина строки в символах, ER - флаг ошибки. При ошибочном задании параметров ER=.TRUE. , при правильном - ER=.FALSE.

Перед каждым объявлением страницы необходимо стирать с экрана остатки прежней.

2. NSNLIN - выдача номера и координат первой свободной строки.

Обращение: CALL NSNLIN (N, IX, IY, NLAST) ,

где N - номер выданной строки, IX , IY - ее координаты, NLAST - флаг переполнения страницы.

Если страница переполнена, NLAST<0. Страница становится переполненной, когда в момент запроса строки все строки на экране оказываются занятыми.

3. NSFIXA - объявление абзаца.

Обращение: CALL NSFIXA

В качестве начальной строки объявляемого абзаца принимается первая свободная строка.

4. NSRETA - возврат абзаца.

Обращение: CALL NSRETA

Указатель начала свободной области страницы переводится в позицию первой строки возвращаемого абзаца.

5. NSRETP- возврат страницы.

Обращение: CALL NSRETP

Указатель начала свободной области страницы переводится на начало страницы.

6. NSGETP- выдача значений параметров страницы и системных переменных.

Обращение: P =NSGETP (T)

Значение параметра T :

Значение функции P :

1HX	- координата X начала первой строки страницы;
1HY	- координата Y начала первой строки страницы;
4HSTEP	- интервал между строками;
4HNLIN	- число строк в странице;
4HLLIN	- число символов в строке;
3HCUR	- номер первой свободной строки;
3HFIX	- номер первой строки последнего абзаца;
4HCURP	- номер последнего абзаца;
3HFRE	- число строк страницы, остающихся свободными;
3HPAR	- число строк в последнем абзаце.

Общая длина всех подпрограмм нижнего уровня составляет 417₈ слов.

ЛИТЕРАТУРА

1. Кавченко А.В. и др. ОИЯИ, 10-9325, Дубна, 1975.
2. Карлов А.А., Кирилов А.С., Смолякова Т.Ф. ОИЯИ, P11-12160, Дубна, 1979.
3. Карлов А.А., Кирилов А.С. ОИЯИ, P11-12133, Дубна, 1979.
4. Карлов А.А., Смолякова Т.Ф. ОИЯИ, P11-10440, Дубна, 1977.
5. Карлов А.А., Кирилов А.С., Полинцев А.Д., Смолякова Т.Ф., ОИЯИ, Б1-11-11999, Дубна, 1978.
6. Карлов А.А., Смолякова Т.Ф. ОИЯИ, Д10,11-11264, Дубна, 1978, с.485-493.

Рукопись поступила в издательский отдел
12 марта 1980 года.