



Ордена Ленина и ордена Октябрьской Революции

Институт атомной энергии

им. И. В. Курчатова

А.Д. Сидоренко

ИАЭ-3802/16

**CLUST — ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ
ДЛЯ РЕШЕНИЯ ЗАДАЧ
РАДИАЦИОННОЙ ФИЗИКИ ТВЕРДОГО ТЕЛА**

Москва 1983

ГУБРИКАТОР ПРЕПРИНТОВ ИАЭ

1. Общая теоретическая и математическая физика
2. Ядерная физика
3. Общие проблемы ядерной энергетики
4. Физика и техника ядерных реакторов
5. Методы и программы расчета ядерных реакторов
6. Теоретическая физика плазмы
7. Экспериментальная физика плазмы и управляемый термоядерный синтез
8. Проблемы термоядерного реактора
9. Физика конденсированного состояния вещества
10. Физика низких температур и техническая сверхпроводимость
11. Радиационная физика твердого тела и радиационное материаловедение
12. Атомная и молекулярная физика
13. Химия и химическая технология
14. Приборы и техника эксперимента
15. Автоматизация и методы обработки экспериментальных данных
16. Вычислительная математика и техника

Индекс рубрики дается через дробь после основного номера ИАЭ.

Ордена Ленина и ордена Октябрьской Революции
Институт атомной энергии им. И.В. Курчатова

А.Д. Сидоренко

**CLUST – ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ
ДЛЯ РЕШЕНИЯ ЗАДАЧ
РАДИАЦИОННОЙ ФИЗИКИ ТВЕРДОГО ТЕЛА**

**Москва
1983**

УДК. 519.68

Ключевые слова. точечные дефекты, кластеры, численные методы, дифференциальные уравнения.

Пакет CLUST представляет собой набор программ для решения систем обыкновенных дифференциальных уравнений специального вида с записью результатов счета в архив. Управление процессом счета ведет специальный монитор, что существенно облегчает отладку программ пользователя и повышает эффективность использования машинного времени. Вывод результатов и просмотр архива могут осуществляться на АЦПУ и (или) на терминал. Пакет имеет структуру, позволяющую легко перестраивать его для решения других задач. ~~В настоящей работе~~ описана работа с пакетом на ЭВМ БЭСМ-6 и приведены способы его перестройки.

ВВЕДЕНИЕ

Решение многих задач физики включает в себя численное интегрирование систем обыкновенных дифференциальных уравнений. Если число переменных велико, то результатом вычислений на ЭВМ является обычно не подробная таблица значений всех переменных на каждом шаге, а временные зависимости неких функций этих переменных и подробная информация о решаемой системе в заранее заданные моменты времени. Кроме того, нас интересует возможность записи результатов счета в архив с последующим продолжением счета и (или) выводом результатов на печать (или график).

Аналогичные требования предъявляются к программам для решения дифференциальных уравнений в частных производных от двух переменных. В этом случае требуется еще обычно выводить на график функцию распределения.

Для решения подобных задач в библиотеки стандартных программ общего пользования записываются модули, обеспечивающие выполнение одного шага интегрирования. Такие модули обычно плохо защищены от ошибок в программах пользователя, что делает написание эффективных программ и их быструю отладку возможным лишь для простых случаев. Квалифицированные пользователи, обращающие особое внимание на высокое качество программ, предпочитают писать эти блоки сами, а не использовать готовое обеспечение.

В данной работе предпринята попытка написания пакета прикладных программ, обеспечивающего эффективную работу при счете сложных задач, легкую замену отдельных модулей и максимальную экономию памяти. Все операции вывода результатов на печать, графики и выдача отладочных сообщений могут инициироваться автоматически по окончании счета или отдельным заданием.

Важной задачей в радиационной физике твердого тела является задача о кинетике кластеров точечных дефектов в металле под облучением, имеющая приложения для проблемы управляемых термоядерных реакторов и реакторов на быстрых нейтронах [1 – 3]. При ее решении мы сталкиваемся со значительными математическими трудностями, связанными с требованием максимально возможной эффективности программы из-за ограничений по памяти и скорости счета [4]. Для решения этой задачи был создан пакет прикладных программ CLUST. Возможна сравнительно легкая переделка отдельных блоков пакета для решения других задач с общим числом переменных не более 500. Так, он использовался для расчета характеристик химического лазера [5]. В данной работе описывается общая структура пакета и обращение к нему для решения системы уравнений, описывающих зарождение и рост дислокационных петель и вакансионных пор в металле, облучаемом потоком быстрых частиц.

1. СТРУКТУРА ПАКЕТА

Вся работа с комплексом программ CLUST состоит из формирования заданий для него. Существует три основных типа заданий:

- 1) NEWR — задание на запись информации о новом варианте в архив;
- 2) JUMP — задание на счет;
- 3) PRTA — задание на вывод результатов и (или) распечатку справочной информации по архиву.

Любое задание может быть инициировано из пакета или с помощью диалогового монитора системы, являющегося одним из блоков многофункциональной диалоговой системы SERVIS, работа с ним будет подробно описана в следующей статье. В данной работе описывается выдача заданий для CLUST в пакетном режиме. В каждом разделе сначала приводится описание общей структуры соответствующего задания, потом дается текст для его запуска на ЭВМ БЭСМ-6.

1.1. Запись нового варианта в архив

В качестве архива используется канал прямого доступа на МД или МЛ [6]. Идентификатором данного варианта является константа ISO из трех символов, например:

NAME = ЗНП45,

NAME = ЗНТТТ.

Все записи в канале ПД, относящиеся к данному варианту, начинаются (по имени) с этих трех символов. Каждому варианту соответствуют, по крайней мере, три записи:

1) справочная информация о варианте, набор констант для генерации коэффициентов системы уравнений и т.д.;

2) значения всех переменных в некий момент времени и отладочная информация для этого момента времени;

3) значения величин, временную зависимость которых мы хотели бы иметь для вывода на график.

Кроме этих записей, к данному варианту могут относиться еще файлы той же структуры, что и второго типа для других моментов времени. Опишем подробно назначение каждого из файлов.

Имя файла первого типа образуется конкатенацией идентификатора варианта и константы ЗН_μS, например:

NAME = 6НП45_μS,

NAME = 6НТТТ_μS.

В конфигурации, использованной нами для счета, в этом файле была записана следующая информация:

краткая справка о варианте (печатается всегда);

подробное описание варианта (печатается по требованию);

параметры для генерации структуры уравнений;

параметры для генерации точек записи информации в архив;

параметры генерации будильников для периодического сброса результатов счета на МД с целью сохранения результатов счета при сбоях ЭВМ.

Приведем фрагмент пакета перфокарт, обеспечивающих запись:

*

*FOREX

*PERSO: 42060

BLOCK DATA

COMMON/NAME/INA

DATA INA/ЗНП45/

END

BLOCK DATA

COMMON/DCYCLE/DELT, AKD, TED, TH, AKH,

TEH, TL, AKL, TEL, NPT, TMAX

DATA TH/1.E3/, TL/5.E6/

DATA TEH/673./, TEL/673./

```

DATA AKH/1.E-6/,AKL/1.E-6/
DATA NPT/500/,TMAX/1.E7/
COMMON/FILER/TR,PVAR,PTOT,KEDGE,TE,
      AKR,NIV,NIL,N
DATA N/500/,NIV/200/
DATA KEDGE/20/
END

```

```

BLOCK DATA
COMMON/HICRO/HV,HI,FRV,FRI,A,OMEGA,EI,
      FI,FV,RD
* ,DI,DV,PREFI,PREFV,ALPHA,OI,OV,ZI,
      ZV,EJ,BETA

```

```

DOUBLE PRECISION OMEGA
DATA EV,EI/1.6,4./
DATA RD/0.0/
DATA ALPHA/1.E3/
DATA PREFI,PREFV/1.02,1./
DATA ZI/1.02/,ZV/1.00/
DATA EJ/0.8/,BETA/5./
DATA HV,HI/1.3,.15/
* ,FRV,FRI/1.E13,1.E14/
* ,A/3.63E-8/
* ,OMEGA/1.0-23/

```

```

END
BLOCK DATA
COMMON/EDREC/DST,ST(10)
DATA NST/2/
DATA ST/1.E3,1.E4/
END

```

```

*CALL NEWP
*CALL NEWSOR
  СЧЕТ С НОВЫМИ ПРЕФЕРЕНСАМИ
*FC

```

где EI, EV — энергия образования междоузельных атомов и вакансий соответственно; RD — плотность линейных дислокаций; ALPHA — коэффициент поверхностного натяжения; HI и HV — энергии миграции; FRI и FRV — частоты для междоузельных атомов и вакансий соответственно; A — постоянная решетки; OMEGA — атомный объем. Параметры ZI, ZV, PREFI, PREFV описывают предпочтения стока на линии.

ные дислокации и петли. Параметры DTRT и DTS задают интервалы процессорного времени: каждые DTRT с обновляется информация для продолжения счета при следующем запуске, каждые DTS с в архив записывается файл структуры второго типа. Значение остальных параметров описывается вместе с соответствующими программами.

Имя файла второго типа образуется из имени варианта и константы вида $3N_{i}R$, например:

$NAME = 6HP45_{i}R$,

$NAME = 6HTT_{i}R$.

Файл с таким именем отражает состояние интегрируемой системы уравнений в некоторый момент времени. Этот файл используется для возобновления счета при очередном формировании задания на счет (если этот вариант в предыдущий раз не был досчитан до конца). Кроме этого, файл второго типа используется для записи подробной информации в заранее заданный момент времени. При этом имя файла образуется следующим образом: если TR есть текущее время системы, а TMAX есть верхний предел интегрирования, то вычисляется величина $NOM = TR \times 1000 / TMAX$, затем преобразуется по формату I3 и используется для формирования имени, например:

$TR = 200.16$,

$TMAX = 1.E5$, тогда

$NOM = 2$, $NAME = 6HTT_{i}2$.

Опишем содержимое файла второго типа:

текущее время системы уравнений;

процессорное время, затраченное на этот вариант за последний запуск на счет;

полное процессорное время, затраченное на этот вариант за все запуски на счет;

значения всех переменных решаемой системы уравнений;

информация о структуре уравнений;

некоторые из коэффициентов системы уравнений, представляющие наибольший интерес при анализе решения.

При записи (чтении) этого файла мы пользуемся COMMON/FILER/TR, PVAR, PTOT, KEDGE, TE, AKR, N1V, N1L, N, RY (502).

Здесь TR — текущее время системы; PVAR, PTOT — времена, затраченные на счет этого варианта, как описано выше; KEDGE — число дефектов в кластере, начиная с которого происходит переход на групповой метод (см. описание программы FCOEF); TE — текущая

температура; AKR — текущий коэффициент генерации точечных дефектов; N1V, N1L и N — число уравнений для пор, петель и полное, RY — переменные.

Имя файла третьего типа образуется из имени варианта и константы вида 3HNLТ, например:

```
NAME = 6HP45NLТ,
```

```
NAME = 6HTTTLТ.
```

В этот файл заносится информация о временных зависимостях. Для считывания используется COMMON/FILET/NLT, NMAX, ТВ (4000).

Здесь NLT — число точек записи по времени; NMAX — максимальное число точек записи. Требуемые записи располагаются в массиве ТВ, каждая занимает NMAX ячеек.

1.2. Запуск задания на счет

Для запуска задания на счет необходимо, чтобы информация о считаемых вариантах была уже записана в архив. Так как запись в архив нового варианта проводится из функциональной части пакета, до карты x EXECUTE, то возможна одним запуском инициация нового варианта и выход его на счет. Для отладочных целей предусмотрена возможность запуска на счет без подключения архива. В этом случае составляем пакет так же, как при записи нового варианта, но без занесения справочной информации, а перед вызовом программы NEWR подкладываем дополнительно:

```
BLOCK DATA  
COMMON/REGIM/IPULT  
DATA IPULT/5HNQMD/  
END
```

Приведем фрагмент управляющих карт для выхода на счет:

```
*FOREX  
*PERSO:42000  
*PERSO:42000,CONT  
BLOCK DATA  
COMMON/NAME/INA  
DATA INA/3HLV4/  
END
```

```
*MAIN JUMP
*CALL FICMEMORY
*EXECUTE
*CALL PRTA:CAT
*CALL PRTA
```

Этими картами мы сформировали задание на счет варианта с идентификатором LV4. Если это имя отсутствует в архиве, то будет просмотрен весь архив. В случае обнаружения варианта, не досчитанного до конца, программа интерпретирует это как задание на его счет. После окончания счета, если не истекло заказанное время, будет произведен переход на следующий вариант (при его наличии). Таким образом, все задание на счет может состоять только из вызова головной программы JUMP, без какой-либо дополнительной информации. Признаки того, что данный вариант досчитан или не досчитан до конца, могут проставляться программой ПРИЗНАК. Обращение к ней производится из функциональной части пакета:

```
* CALL ПРИЗНАК:ИМЯ,Е
* CALL ПРИЗНАК:ИМЯ
```

В первом случае будет проставлен, во втором — снят признак для варианта с идентификатором ИМЯ.

1.3. Задание на вывод результатов

Каждое задание на вывод состоит из обращения к программе PRTA из функциональной части пакета:

```
* CALL PRTA: вид работы перфокарты с идентификаторами
* EC (признак конца информации)
```

Вид работы задается 6-буквенными идентификаторами. Имеются следующие виды работы:

CAT — печать каталога архива. Печатаются идентификаторы вариантов и краткая справка о них;

S, R, T или число — печать таблицы о содержимом файлов соответствующей структуры;

G1 — вывод на график функций распределения петель и пор по размеру;

GM — вывод на график временных зависимостей величин.

В случае отсутствия карт за вызывающей программой (но при этом все равно обязателен признак конца) информация распечатывается о первом из недосчитанных до конца вариантов. В противном случае печатается подробная информация обо всех вариантах, идентификаторы которых встретились в списке.

Мы описали общий принцип обращения к пакету прикладных программ CLUST. Перейдем теперь к описанию основных блоков пакета и возможному способу их замены для решения других задач.

1.4. Основные блоки пакета

Основными блоками пакета являются следующие:
генератор сеток GENSET;
генератор точек записи временных зависимостей GENTT;
генератор верхних пределов интегрирования GENB;
генератор коэффициентов системы GENCO;
генератор будильников;
блок работы с записями в канале ПД.

Результатом работы генератора сеток являются массивы, описывающие связь между исходной задачей и задачей численного интегрирования. Так, для уравнения в частных производных этот массив — сетка, на которой мы записываем разностные уравнения, а для задачи о кинетике кластеров точечных дефектов это точки, в которых берутся группы [3]. Для включения в пакет программа-генератор сеток должна отвечать следующим условиям:

- 1) не иметь формальных параметров;
- 2) результатом работы программы является массив действительных чисел в рабочем COMMON/USER/.

Организация входных параметров и согласованность их с другими программами может быть выполнена через общие блоки произвольным способом. В нашем случае для этого используется COMMON /FILER/, а входным параметром является KEDGE, причем генерируется сетка

$$L_0 = \text{KEDGE}, \quad L_{k+1} = L_k (1 + 1/\text{KEDGE}), \quad k \geq 0.$$

Для организации записи временных зависимостей должны быть указаны точки, в которых необходимо производить запись. В связи с требованием экономии памяти, невозможно хранить все требуемые для организации записи массивы в памяти. Так как хранить их на МД нецелесообразно из-за уменьшения скорости счета, то было выбрано представление массивов их генераторами. Этот способ хранения используется, например, в языке ПОПЛАН [7] и эквивалентен замене простого списка динамическим. Для ускорения выборки требуемого элемента массива мы используем промежуточную буферизацию, так как обычно обращение проводится поочередно к близким элементам.

В нашей задаче при расчетах исследовался как непрерывный, так и циклический режим облучения. В связи с тем, что требовалось распределить точки записи по всем циклам равномерно и иметь больше точек вблизи момента переключения режима (для исследования переходного процесса), наш генератор был привязан к точкам переключения режима. Входная информация при этом бралась из

COMMON/DCYCLE/DELT, AKD, TED, TH, AKH, TЕН, TL, AKL,
TEL, NPT, TMAX,

где DELT — время начала первого цикла; AKD и TED — скорость генерации точечных дефектов и температура до начала циклического режима; TЕН и TEL — температуры во время циклов; AKH и AKL — скорости генерации; TH и TL — продолжительности циклов. Весь расчет ведется до момента времени TMAX, а число точек записи при этом равно NPT.

Для организации записи значений всех переменных в заданные моменты времени и переключения режимов при резкой смене коэффициентов системы используется следующая процедура: из всех точек времени, в которых нам нужно будет проделывать такие операции, выбирается наименьшая справа от текущего времени системы и верхний предел интегрирования полагается равным ей. Тогда по достижении этой точки управление передается в программу GENB, которая выполняет нужные действия и полагает предел интегрирования равным следующему значению. Генерация этих точек в нашем случае осуществляется программой SEPDIF. Задание точек записи производится в COMMON/FDREC/NST, ST (10), где NST — число

точек записи, а массив ST служит для их явного указания. Для того чтобы программа-генератор точек верхних пределов могла быть включена в комплекс без дополнительной настройки, она должна удовлетворять следующим требованиям:

1) обращение к ней CALL GENB (A, B), где A -- текущий момент времени; B -- очередной предел интегрирования;

2) все остальные требуемые параметры передаются через общие блоки.

Под генератором коэффициентов системы мы понимаем программы, обеспечивающие задание всей необходимой информации об уравнениях, требуемой для выполнения одного шага интегрирования. В нашей работе использовались программы SOLVER [8], для которых мы формировали следующую информацию:

1) описание правых частей системы уравнений;

2) описание расположения ненулевых элементов якобиевой матрицы системы. Задание этого блока производится в произвольной форме, так как он должен быть согласован с используемым методом интегрирования.

Для организации периодического сброса информации на МД, сбора статистики по времени центрального процессора, используемого отдельными блоками и для организации отладки используется ряд "будильников". Будильник для записи данных в канал ПД устроен следующим образом: после выполнения каждого шага интегрирования мы анализируем процессорное время, сравнивая его с временем, когда было проведена запись в предыдущий раз. Как только разность этих времен становится больше заданной величины, мы записываем на МД интересующие нас переменные. Будильник второго типа устанавливается заданием прерывания "истекло время по ЭК" и используется для распечатки диагностической информации в аварийных ситуациях. Будильник третьего типа работает при помощи аппарата событий и используется в отладочных целях для измерения времени работы отдельных подпрограмм.

Стандартные возможности работы с каналом прямого доступа предусматривают чтение (запись) файлов по их имени в каталоге. В связи с тем, что отводить по отдельному файлу для каждой интересующей нас группы величин нецелесообразно (файл в канале ПД не может быть меньше одной зоны, а это приведет к неэкономному расходу зон на МД), были разработаны дополнительные программы

для организации в пределах одной зоны нескольких записей. Подробное описание этих программ будет дано в следующем разделе. Они включают в себя программы "нижнего" и "верхнего" уровней, т.е. имеется возможность как использования готовых программ для организации обмена с внешними носителями, так и модификация их применительно к нуждам пользователя.

2. ПОДРОБНОЕ ОПИСАНИЕ ПРОГРАММ ПАКЕТА

В этом разделе мы приведем подробное описание программ пакета CLUST. В качестве стандартного метода интегрирования используется метод Гира. Вся настройка на конкретно используемые методы вынесена в отдельные подпрограммы, что позволяет оперативно менять их, не затрагивая собственно программ пользователя. Предусмотрено занесение в архив результатов счета подробной информации об именах всех подпрограмм, которые необходимо использовать для обработки решения, что позволяет хранить оттранслированные модули в одной библиотеке, а запуск заданий на счет и обработку решения выполнять единообразным по структуре пакетом перфокарт.

2.1. Головная программа выхода на счет

Головной программой выхода на счет является программа JUMP. Для ее работы требуется следующая информация, предварительно записанная в архив:

- 1) имя программы, осуществляющей настройку констант и заказывающей рабочую память для программ интегрирования, — JSOLV;
- 2) имя программы, осуществляющей чтение информации из архива и начальное заполнение рабочих ячеек/типа установки будильников/;
- 3) имя программы, передающей управление программе интегрирования, — JEXEC.

Выполнение программы начинается с вызова монитора пакета CLMON, который управляет прохождением всего счета. Далее ана-

лизируется содержимое COMMON/REGIM/. Затем осуществляется считывание из архива имен, записанных в файл с именем, оканчивающимся на S, и заполняется COMMON/JPNAM/JSOLV, JISET, JEXEC с именами служебных программ. В нашем случае ISOLV = 6HEREMIN и настройка проводится на пакет SOLVER [8]. Написание этой программы не представляет каких-либо сложностей и основывается на инструкции к соответствующей программе для интегрирования. JSOLV загружается динамически, резервирует при этом необходимую память. Вызов последующих подпрограмм производится из нее.

Программа JISET начинает свою работу со считывания информации из архива. Вся считываемая информация размещается в COMMON-блоках, что позволяет, в принципе, вести работу и без архива, засылая ее каким-либо другим способом. Пользователю поставляется программа с именем SSSSET, выполняющая следующие действия:

- 1) заполняет COMMON/FDREC/ — для генератора верхних пределов;
 - 2) заполняет COMMON/CCLOCK/ — для генератора будильников;
 - 3) заполняет COMMON/SSTIME/ — для генератора точек записи;
 - 4) осуществляет начальную установку будильников, шага интегрирования и буферов обмена;
- Б) заполняет рабочие ячейки служебных программ и определяет режим, в котором будет находиться решаемая система.

Далее вызывается программа JEXEC, передающая управление программе-интегратору. Отметим, что многие библиотечные программы интегрирования рассчитаны на выполнение всего одного шага. В этом случае пользователь должен предусмотреть в JEXEC многократное обращение к вызываемому модулю, с выходом на каждом шаге на обработку решения. Информация для генераторов передается в виде содержимого общих блоков, имеющих структуру, описанную ниже. При работе всех программ предусмотрена печать отладочных сообщений. Управление печатью осуществляется через параметр MGSLEV в общем блоке COMMON/REGIM/, который может принимать значения 0, 1, 2. Значение 0 выключает печать сообщений, кроме аварийных, а 1 и 2 соответствуют печати краткой и подробной диагностики. Этот же параметр передается в пакет SOLVER, где он вы-

полняет аналогичную функцию. Кроме того, MGSLEV = 2 включает печать аварийных распечаток (АВРАСП), которые в остальных случаях блокируются.

2.2. Генераторы точек переключения режима и верхних пределов

Стандартный генератор в качестве верхнего предела интегрирования может выбрать одну из следующих точек:

- точку, в которой необходимо произвести запись на МД;
- точку переключения режима;
- точку "сброса" времени.

При проведении интегрирования до больших времен (по сравнению с минимальным шагом интегрирования) желательно периодически перепределять текущее время системы, особенно при наличии точек скачкообразного переключения режима. Реальное время при этом хранится в COMMON/FILER/. Управление генератором идет через два общих блока, /FDREC/ и /DCYCLE/.

COMMON/FDREC/NST, ST (10)

где TSBR — интервал, через который будет производиться "сброс" времени, а остальные параметры уже были описаны в разд. 1.1. Если этот интервал задан равным 0 (или вообще не задан), то сброса происходить не будет.

2.3. Установка будильников

Для установки будильников служит общий блок:

COMMON/CCLOCK/DTRT, DTS, AM (3),

где параметры DTRT и DTS описаны в разд. 1.1.

2.4. Работа с записями в канале ПД

Для легкой перестройки программы с одного метода на другой и использования различных структур данных удобно все записи в архиве идентифицировать по имени варианта и имени подпрограммы, в которой они используются. При этом многие записи имеют небольшую длину по сравнению с минимальным объемом записи в канале прямого доступа (даже при записи одного числа в канал ПД ему соответствует целая зона). Поэтому использование для каждой записи своего идентификатора привело бы к крайне неэкономному использованию места на МД. Для обеспечения работы с подобными "короткими" файлами был написан ряд программ. Опишем сначала программы "нижнего" уровня.

Для работы с несколькими записями в одном файле ПД используется следующая идеология: в памяти задачи выделяется буфер, на котором формируется необходимая сложная запись, далее вся работа проводится с ним. Длина буфера определяется при этом суммарной длиной всех размещаемых в нем рекордов плюс число записей, умноженное на два. При обращении к соответствующей программе производится запись или чтение из этого буфера в указанное место памяти. Кроме этого, используется массив из семи элементов, в которых задается управляющая информация.

АДРЕС = ADDRESS (M)

Переменной АДРЕС присваивается значение адреса переменной M в памяти. Эта программа используется, если в качестве буфера берется массив, адрес которого неизвестен.

Каждая запись в буфере идентифицируется двумя именами, причем первое из них 3-буквенное, а второе 6-буквенное. В качестве имен могут также использоваться любые восьмеричные числа, причем для первого имени значащими являются лишь старшие восемь цифр (24 разряда). Структура управляющего массива:

- 1) первое имя записи;
- 2) второе имя записи;
- 3) начало буфера;
- 4) длина буфера;
- 5) начало области памяти под запись;

- 6) число слов в записи;
- 7) рабочая ячейка (пользователю менять нельзя).

KO = LBDNEW (IN)

Начало работы с новым буфером. Перед обращением необходимо занести в управляющий массив адрес начала буфера и его длину, при этом максимальная длина равна 7777В. Если все в порядке, то KO = 0В.

KO = LBDWRI (IN)

Запись переписывается в буфер. Перед обращением задать имена записи, ее длину, местоположение в памяти и параметры буфера.

Код ответа:

- 0В – все в порядке;
- 1В – запись с таким именем уже есть;
- 2В – контроль каталога (нет контрольных слов в буфере);
- 3В – контроль записи;
- 4В – нет места под запись такой длины.

П Р И М Е Р

Пусть мы хотим использовать в качестве буфера массив А, причем не с начала, а с 1000 элемента. Тогда следующая последовательность операторов:

DIMENSION A (10 000), B (100), UPR (7)

UPR (3) = ADDRESS (A (1000))

UPR (4) = 7000

UPR (5) = ADDRESS (B)

UPR (1) = 'TEC'

UPR (2) = 'TECTN1'

UPR (6) = 20

C = LBDNEW (UPR)

C = LBDWRI (UPR)

вызовет запись 20 элементов В в буфер под именами TEC и TECTN1.

KO = LBDREA (IN)

Запись считывается из буфера. Коды ответа те же, что для LBDWRI кроме KO = 1B, означающего отсутствие такой записи.

KO = LBDDEL (IN)

Запись уничтожается в буфере. Коды ответа те же, что в предыдущем случае.

KO = LBDINF (IN)

На входе: адрес буфера. На выходе получаем запись из трех элементов: максимальная длина, свободно, число записей. Используется для контроля за величиной свободного места. Все величины выдаются в виде восьмеричных чисел.

При работе в пакете CLUST эти программы используются в совокупности с программами работы с каналом прямого доступа. Канал прямого доступа при этом используется для записи/считывания содержимого буферов. Для конкретной реализации были выбраны следующие параметры:

- 1) буфер — первые 2000 элементов COMMON/CORE/;
- 2) память под запись — COMMON/USER/;
- 3) первое имя — совпадает с последними тремя символами имени файла в каталоге канала;
- 4) второе имя — обычно имя подпрограммы, использующей эту информацию.

Максимальная длина записи при этом равна 1000. Буферная память используется также рядом других программ, поэтому перед ее использованием она сбрасывается на МБ, а после восстанавливается. При записи/чтении используются две программы, использующие работу с буферами — CLREA и CLWRI.

CALL CLWRI (имя1, имя2, N)

Производится запись в канал ПД N элементов из COMMON/USER/. Запись идет в файл ПД с именем, образуемым из имени варианта и пер-

вым именем записи. Следить за тем, был ли файл с таким именем, нужно ли открывать новый буфер и т.д., будет сама программа.

CALL CLREA (имя1, имя2, N)

Будет считана из канала ПД требуемая информация, ее длина засылается в N.

Применение описанных выше программ требует дополнительно 210В ячеек памяти по сравнению с работой с "чистым ПД". Кроме того, увеличивается число обменов. Однако при этом почти вся память используется для размещения программ, а не для многочисленных рабочих массивов. При решении сложных задач выполнение одного шага интегрирования занимает время порядка нескольких секунд (или даже десятков секунд). При этом увеличение времени счета за счет дополнительных обменов несущественно. При времени простых систем, когда время выполнения одного шага мало, относительное увеличение времени счета будет весьма значительным только в том случае, когда число точек чтения (записи) очень велико. Таким образом, описанные выше программы можно использовать практически во всех случаях.

При работе из ФОРТРАНа могут возникнуть ошибки при засылке констант. Так, после выполнения оператора

A = 0В

перед засылкой в ячейку A будет проведено преобразование типа выражения в правой части. В результате в A окажется не восьмеричный 0, а 4000 0000 0000 0000В – результат преобразования к "вещественному" типу. Для того чтобы избежать подобных ошибок, рекомендуется использовать п/п TOTON и TOTOC. После вызова

CALL TOTON (ОТКУДА, КУДА)

содержимое ячейки памяти, соответствующей идентификатору ОТКУДА, будет без каких-либо изменений переписано в ячейку КУДА. Обращение к п/п TOTOC проводится аналогично, но содержимое пересылается с командной сверткой. Эти же программы используются при рассылке элементов из массивов, в которые было произведено считывание из буфера, по необходимым адресам.

Отметим также, что время записи (считывания) в буфер невелико — на поиск начала информации уходит в среднем 10 мкс. Поэтому в случае не очень частого обращения может оказаться выгодным хранить в таком буфере различные массивы (при этом мы получаем "динамическое" распределение памяти).

ВЫВОДЫ

1. Возможно построение комплекса программ для решения систем обыкновенных дифференциальных уравнений с записью промежуточных результатов счета, легко настраиваемого на конкретно требуемую структуру записей.

2. Для выбора оптимального метода решения подобных задач целесообразно организовывать архив и программы таким образом, что запуск различных вариантов осуществляется единообразно.

3. Запись входной информации для данного варианта и запуск его на счет должен осуществляться отдельными заданиями, что существенно облегчает процесс отладки и поиска ошибок.

4. Написан монитор, обеспечивающий управление заданиями на счете, при помощи которого возможно быстрое нахождение ошибок в программах пользователя.

5. Построенный комплекс программ обеспечивает эффективную работу программиста невысокой квалификации.

Автор выражает признательность А.Ю. Еремину (ВЦ АН СССР) за помощь в работе с пакетом программ SOLVER и Ю.Д. Лизунову за ценные обсуждения.

Список литературы

1. Kiritani M. — Journ. Phys. Soc. Japan, 1973, vol. 35, p. 95.
2. Koiwa M. — Ibidem, 1974, vol. 37, p. 1532.
3. Hayns M.R. — Journ. Nucl. Mat., 1976, vol. 59, p. 175.
4. Lam Nghi Q. and Johnson R.A. — Nuclear Metallurgy, 1976, vol 20, p. 121.

5. Никеров В.А., Русанов В.Д., Сидоренко А.Д., Шолин Г.В. – В кн.: Вопросы атомной науки и техники. Сер. Термоядерный синтез, 1980, вып. 1 (5), с. 115.
6. Веретенев В.Ю. Работа с внешней памятью, как с устройством прямого доступа в мониторной системе "Дубна". Информатор № 5, 1974, М., ИАЭ.
7. Баяковский Ю.М., Вьюкова Н.И., Галатенко В.А., Ходулев А.Б. Программирование на языке ПОПЛАН. М.: ИПМ АН СССР, 1976.
8. Еремин А.Ю., Марьяшкин Н.Я. Пакет программ SOLVER. Часть 1. Системы нелинейных алгебраических и обыкновенных дифференциальных уравнений в разреженными якобиевыми матрицами. М.: ВЦ АН СССР, 1980.

Редактор Г.Я. Кармодонова
Технический редактор Н.А. Малькова

Т-06419. 22.02.83. Формат 60x90/16. Уч.-издл. 1.1
Тираж 300. Индекс 3624. Заказ 1296

Отпечатано в ИАЭ

Препринт ИАЭ-3802/16. М., 1983