СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

10 - 8556

Х.Кэниг, Л.С.Нефедьева

ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛАМИ В СИСТЕМЕ ОБРАБОТКИ СПЕКТРОВ (СОС)

1975

10 - 8556

Х.Кэниг, Л.С.Нефедьева

ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛАМИ В СИСТЕМЕ ОБРАБОТКИ СПЕКТРОВ (СОС)

Современный уровень развития экспериментальных методов ядерной физики характеризуется большим потоком информации, который в большинстве случаев не может быть обработан без применения современной вычислительной техники. Эффективность использования ЭВМ достигается путем создания математических систем, позволяющих автоматизировать процесс приема, накопления и обработки потока исходной информации /I/. От вида информации зависит построение математических систем, которые должны быть сориентированы в определенной области. Поток спектрометрической информации характеризуется набором массивов данных строго определенной структуры. Поэтому при разработке проблемно-ориентированной системы обработки спектров (СОС) на ЭВМ БЭСМ-6 важное место занимает организация работы с массивами данных /2/. Любая последовательность данных, которая передается системе для хранения и обработки, называется массивом данных. Допускаются различные типы формата массивов данных:

Тип I - спектр ,

Тип 2 - набор физических параметров ,

Тип 3 - любая числовая информация ,

Тип 4 - алфавитно-цифровая информация.

Введение понятия формата массива обусловлено программами обработки, но оно не имеет существенного значения для принципов организации работы с массивами данных.

Эта часть СОС опрецеляет работу других частей системы. С другой стороны,она сама сравнительно самостоятельна.

Основные задачи организации работы с массивами данных:

I. Хранение массивов данных на личных носителях. Под личными носителями понимаются перфокарты и магнитные ленты БЭСМ-6, СДС-608 и ЕС-5012.

2. Быстрый доступ к массивам.

3. Обмен массивами между программными модулями.

4. Обмен массивами между системой СОС и операционной системой БЭСМ-6.

5. Объединение массивов данных в общий массив, что позволяет организовать работу оператора циклов с массивами.

Для решения этих задач было введено понятие"файл".

I. Описание файлов

Под файлом понимается символическое изображение массива данных,[3] которое однозначно определяет его в системе СОС . Каждому файлу сопоставляется имя, выраженное идентификатором. Файл может состоять из любого числа подфайлов, которые логически не связаны друг с другом. Подфайл - это логическая единица данных, снабженная именем файла, к которому она относится, и номером подфайла в данном файле. Один подфайл не должен содержать больше 4096 слов.

Итак, любой массив данных, снабженный однозначным для системы именем, представляет собой файл или подфайл. Внутренняя структура этих массивов не влияет на структуру файла.

Различаются два вида файлов: постоянные файлы - для длительного хранения информации, и рабочие файлы - для хранения промежуточной информации.

I. Постоянные файлы

Носителями хранения постоянных файлов служат перфокарты и магнитные ленты. Для осуществления быстрого доступа к файлам, которые находятся на магнитных лентах, создается постоянный каталог, содержащий информацию, необходимую для работы с магнитными лентами и файлами. Каталог состоит из оглавления и паспортов магнитных лент. Оглавление содержит необходимую информацию о каждом паспорте:

а) имя магнитной ленты;

б) номер бобины магнитной ленты;

в) признак заполнения магнитной ленты;

г) адрес паспорта в каталоге;

д) дату последнего изменения паспорта (если в течение полугода паспорт не изменяется, то он снимается с регистрации).

Паспорт данной магнитной ленты содержит информацию о файлах, записанных на этой ленте. О каждом файле или подфайле имеется в паспорте следующая информация:

а) имя файла;

б) номер подфайла (если он есть);

в) длина файла или подфайла;

г) признак формата.

Помимо того, что паспорт хранится в каталоге, его дубль хранится на соответствующей магнитной ленте. Это позволяет избежать его потери в случае разрушения каталога. Паспорта составляются во время записи файлов на магнитную ленту. Для этого на некоторых машинах ОИЯИ создаются специальные программы, которые подготавливают магнитные ленты по правилам СОС.

Постоянный каталог находится на магнитной ленте. Оглавление и паспорта требуемых магнитных лент система СОС в начале

своей работы передает на магнитный барабан. При этом собирается таблица всех имен файлов, используемых в конкретном задании. Исходя из таблицы имен, паспортов и оглавления составляется временный каталог (аналог постоянного каталога) в МОЗУ. В конце выполнения задания система вносит изменения в соответствующие паспорта и записывает их на магнитные ленты. Таким образом, экономится число обращений к магнитным барабанам в процессе выполнения задания. Для работы с постоянным каталогом создается группа служебных программ, позволяющая выдать на печать информацию из оглавления и паспортов. В случае необходимости можно создать паспорт магнитной ленты.

2. Рабочие файлы

Рабочие файлы существуют в пределах одного задания. Носителями хранения рабочих файлов служат магнитные барабаны.

Рабочие файлы могут быть использованы:

I. Для временного хранения массивов данных. Рабочие файлы дают возможность эффективно организовать обмен массивами данных между программными модулями. Магнитные барабаны предназначены для хранения промежуточных результатов обработки, благодаря сравнительно быстрому доступу к ним. Пользователь имеет возможность всегда переписать рабочий файл или подфайл в постоянный файл.

2.Для составления новых массивов. Рабочие файлы могут служить местом сборки отдельных числовых данных или наборов данных, из которых пользователь может составить новый массив. Для этого вводится понятие бесструктурного рабочего файла. Такой файл не имеет подфайлов. Он представляет собой зарезервированное на магнитном барабане поле, длина которого не должна превышать 4096 слов. Бесструктурный рабочий файл можно всегда переписать в постоянный файл.

Для хранения рабочих файлов имеется память 50К слов на магнитных барабанах. Чтобы обеспечить быстрый доступ к рабочим файлам, вводится буферная память. Она занимает 5К слов в МОЗУ. Обращение к магнитным барабанам всегда идет через буферную память. Таким образом, в МОЗУ находятся те файлы,к которым обращались в последнее время. Информация о рабочих файлах регистрируется во временном каталоге.

П. Операторы работы с файлами

Пользователь при составлении задания на языке приказов СОС может пользоваться рядом операторов работы с файлами:

FILE – оператор описания файлов;

CREATE – оператор организации новых файлов;

DELETE – оператор исключения файлов из постоянного каталога;

ADD – оператор составления массивов;

WRITE – оператор обмена файлами.

Операторы FILE и CREATE относятся к группе невыполняемых операторов языка СОС. FILE описывает местонахождение и вид файлов, используемых в данном задании. CREATE дает возможность объединить файлы и подфайлы в новый файл. С помощью оператора DELETE пользователь может исключить из постоянного каталога любой файл или подфайл. Оператор ADD осуществляет запись в бесструктурные рабочие файлы. WRITE реализует обмен между любыми файлами,а также между системой СОС и носителем данных операционной системы БЭСМ-6.

I. Оператор FILE

Оператор описывает файлы, используемые в данном задании. Он должен стоять в начале задания.

В случае описания постоянных файлов данный оператор имеет форму записи:

FILE список имен описываемых файлов (тип файла, носитель хранения).

Допускается 3 типа постоянных файлов: INPUT , OUTPUT и UPDATE.

INPUT используется, когда:

I) Файлы уже зарегистрированы в постоянном каталоге и имеют защиту записи.

FILE список имен файлов (INPUT , имя магнитной ленты).

Пример:

```
FILE GAMMA, GAM1, GAM2 (INPUT, TAPE01)
```

2) Файлы вводятся с перфокарт.

FILE список имен файлов (INPUT , (формат ввода)).

Пример: CALL CINPUT (F1, F2 ... FN, кол-во, TAPE)

```
FILE GAM3 (INPUT, (10F8.2))
```

OUTPUT используется, когда:

I) Создаются новые постоянные файлы.

FILE список имен файлов (OUTPUT , имя магнитной ленты).

Пример:

```
FILE BETA1, BETA2 (OUTPUT, TAPE02)
```

2) Файлы выводятся на перфокарты.

FILE список имен файлов (OUTPUT (формат вывода)).

Пример: CALL C

```
FILE BETA3 (OUTPUT, (10F8.2))
```

UPDATE используется, когда:

I. файлы уже зарегистрированы в постоянном каталоге и не имеют защиты записи.

FILE список имен файлов (UPDATE , имя магнитной ленты).

Пример: CALL CUPDAT (F1, F2 ... FN, кол-во, TAPE)

```
FILE GAM5 (UPDATE, TAPE01)
```

В случае описания рабочих файлов оператор имеет форму записи:

FILE список имен описываемых файлов (тип файла).

В случае описания бесструктурных рабочих файлов он имеет форму записи:

FILE список имен описываемых файлов (тип файла, длина файла).

Допускается только один тип рабочих файлов WORK , который используется, когда файлы существуют только в пределах одного задания пользователя. WORK действует, как UPDATE , и имеет форму записи:

FILE список имен файлов (WORK).

FILE список имен файлов (WORK , длина файла).

Примеры:

```
FILE HELP, GAM1A, GAM2A (WORK)
FILE CONST (WORK, 2048).
```

Описания типов файлов являются действительными только в пределах одного задания. В каждом задании типы файлов должны быть описаны заново.

2. Оператор CREATE

Оператор объединяет несколько постоянных файлов и подфайлов в один файл под новым именем.

CREATE имя файла (тип файла, имя магнитной ленты)=список имен объединяемых файлов и подфайлов.

Имеются две возможности объединения: постоянное и временное.

I) Постоянное, когда все объединяемые файлы и подфайлы находятся на той же магнитной ленте. Оператор CREATE регистрирует новый файл в постоянном каталоге. При этом уничтожается в каталоге самостоятельность перечисленных файлов и связь подфайлов с их предыдущими файлами. Допускаются только два типа файлов: INPUT и UPDATE . Тип OUTPUT тут бессмыслен, так как оператор CREATE объединяет только зарегистрированные файлы.

CREATE имя файла (INPUT , имя магнитной ленты)=список имен объединяемых файлов и подфайлов.

CREATE имя файла (UPDATE,имя магнитной ленты)=список имен объединяемых файлов и подфайлов.

Примеры:

CREATE GAM(INPUT, TAPE01) = GAMMA, GAM1(1), GAM2(1)

CREATE GAMA(UPDATE,TAPE02) = GAM100(4), GAM101(5)

2) Временное, когда объединяются файлы с различных лент. Для этого вводится новый тип файла - TIME.

CREATE имя файла (TIME) = список имен объединяемых файлов и подфайлов.

Пример:

CREATE GAM200 (TIME) = GAMMA, GAM1, GAM2, GAM100

Файлы этого типа не регистрируются в постоянном каталоге, но регистрируются во временном каталоге. Такой файл существует в пределах данного задания. Тип TIME соответствует типу UPDATE.

Различие между файлами типа TIME и WORK заключается в том, что файлы типа TIME временно объединяют постоянные файлы, которые хранятся на магнитных лентах, тогда как файлы типа WORK являются рабочими файлами.

CREATE относится к группе невыполняемых операторов; его местонахождение обязательно в начале задания. Файлы, описанные в операторе CREATE , не надо описывать в операторе FILE .

3. Оператор DELETE

Оператор исключает файлы или подфайлы, которые находятся на той же магнитной ленте, из постоянного каталога. Исключаемые файлы не надо описывать в операторе FILE.

DELETE список имен файлов или подфайлов(имя магнитной ленты).

Пример:

DELETE GAMMA, BETA(3), PEAK(TAPE01)

4. Оператор ADD

Оператор осуществляет запись в бесструктурный рабочий файл. Запись в такой файл происходит в режиме продолжения.

ADD список имен файлов или подфайлов TO имя бесструктурного рабочего файла.

Примеры:

ADD PEAK, ABC(1),GAM1 TO CONST

ADD DELTA TO CONST

Бесструктурные рабочие файлы можно всегда с помощью оператора WRITE переписать в любой постоянный файл.

5. Оператор WRITE

Оператор осуществляет обмен "файл↔файл" и "файл↔носитель данных операционной системы БЭСМ-6".

В случае обмена "файл↔файл":

WRITE список имен файлов и подфайлов INTO имя файла или подфайла;

WRITE список имен файлов и подфайлов INTO CARD (формат вывода);

WRITE(формат ввода) INTO имя файла или подфайла;

Примеры:

WRITE GAM(1), GAM1(1), GAM2(1) INTO GAMMA

WRITE GAM1(2) INTO CARD (10F8.2)

WRITE CARD (10F8.2) INTO GAM2(2)

CARD является специальным признаком, который указывает обмен с перфокартами.

В случае обмена "файл↔носитель данных операционной системы БЭСМ-6":

WRITE список имен файлов и подфайлов INTO DRUM (адрес)

WRITE DRUM (адрес) INTO имя файла или подфайла

Примеры:

```
WRITE GAM1(2), GAM2(3)  INTO DRUM (30)
WRITE DRUM(40) INTO GAMMA(10)
```

DRUM является специальным признаком, который указывает обмен с магнитным барабаном операционной системы БЭСМ-6. Имеются различия между носителями данных системы СОС и операционной системы БЭСМ-6. Носителями данных в системе СОС являются перфокарты, магнитные ленты, зарегистрированные в постоянном каталоге, и магнитные барабаны для хранения рабочих файлов. Все другие магнитные ленты и барабаны "относятся" к операционной системе БЭСМ-6. Обмен массивами данных между системами осуществляется через определенный магнитный барабан операционной системы БЭСМ-6. Оператор WRITE может передать любой массив данных на этот барабан или наоборот. Этот способ обмена предназначается для тех пользователей, которые хотят временно использовать систему СОС.

Предусматривается обмен между системой СОС и магнитными лентами, которые записаны не по правилам СОС.

Все перечисленные операторы доступны для пользователя. Это означает, что при составлении задания можно использовать любой набор из этих операторов. Таким образом, пользователь должен сам решить вопрос о размещении файлов как постоянных, так и временных.

III. Ввод-вывод файлов в программных модулях

Обмен файлами с программными модулями осуществляется с помощью специальных программ. Обращение к этим программам должно находиться внутри модулей. Автору программных модулей не обязательно знать, где находится файл. Ему достаточно в своем модуле в качестве формального параметра указать имена файлов. При обращении к системе из модуля наряду с именем файла надо указать имена своего рабочего поля, на которое система поставит требуемый файл или с которого отцаст его системе.

I. Программы CGET и CPUT

Программы осуществляют передачу файлов в рабочие поля программных модулей.

Для ввода:

```
CGET (FILE, FIELD, LENGTH, FORMAT)
```

Для вывода:

```
CPUT (FIELD, FILE, LENGTH, FORMAT)
```

где FILE – массив из двух слов, который содержит имя файла и номер подфайла;

FIELD – название рабочего поля программного модуля ;

LENGTH – длина файла или подфайла ;

FORMAT – признак формата (указывается числом).

CGET и CPUT позволяют прямой обмен только между рабочими полями программных модулей и постоянными или же рабочими файлами, зарегистрированными в системе. Не допускается прямой обмен между рабочими полями модулей и носителями данных, которые не относятся к системе СОС. Это можно сделать в задании с помощью оператора WRITE.

Вызов обеих программ происходит через фортрановский оператор CALL .

Пример: Сложение двух спектров

```
      SUBROUTINE SUMME  (SPECT1,SPECT2, RESULT)
      COMMON /COC1/ A(4096) / COC2 / B(4096)
      CALL  CGET (SPECT1, A, LENG1, 1)
      CALL  CGET (SPECT2 , B, LENG2, 1)
      IF (LENG1. LT. LENG2) GOTO  2
      DO 1 J = 1, LENG2
    1 A(J) = A(J ) + B(J)
      CALL   CPUT (A, RESULT, LENG2, 1)
```

```
      RETURN
2     DO  3  J = 1,LENG1
3     B(J) = A(J) + B(J)
      CALL   CPUT  ( B, RESULT, LENG1, 1)
      RETURN
      END
```

Введение именных COMMON -блоков COC1 и COC2 позволяет экономить место под рабочие поля программных модулей.

2. Программа CADD

Программа переписывает отдельные данные или наборы данных в бесструктурный рабочий файл.

CADD (FILE, FIELD, BELOW, ABOVE, FORMAT)

Здесь FILE - имя бесструктурного рабочего файла;

FIELD - массив, который содержит переписываемые данные;

BELOW - указатель начала переписываемых данных в массиве FIELD;

ABOVE - указатель конца переписываемых данных в массиве FIELD;

FORMAT - признак формата (последняя запись в файл определяет его формат).

Программа CADD работает аналогично оператору ADD ,т.е.она дает возможность при многократном обращении к одному и тому же бесструктурному рабочему файлу заполнять его данными в режиме продолжения записи.

Пример: Выделение части спектра

```
      SUBROUTINE SELECT (FILE,BELOW,ABOVE,ADDFIL)
      INTEGER BELOW, ABOVE
      COMMON /COC1/A(4096)
      CALL   CGET (FILE,A,LENG,1)
```



```
      CALL CADD (ADDFIL,A,BELOW,ABOVE,2)
      RETURN
      END
```

IУ. Заключение

Предлагаемый набор операторов и программ, осуществляющих работу с массивами данных, является замкнутым модулем в системе СОС, что позволяет с некоторыми изменениями использовать его в других аналогичных системах. Перечисленный набор операторов реализован на машине БЭСМ-6 и проверен на базе программ предварительной обработки спектров. Язык общения системы СОС и системная часть, связанная с составлением каталогов, находится в стадии отладки.

У. Пример

Рассмотрим пример использования набора операторов работы с файлами после окончания полной реализации системы СОС.

Имеются на магнитной ленте с именем TAPE01 три спектра ABC, AA0001 и AA0002, файл GAM1 и набор параметров под именем PEAK.

Задача:

1. Объединить файлы ABC, AA0001 и AA0002 в новый постоянный файл GAM2 .

2. Умножить спектры в GAM1 на 0.334 по программному модулю CMULT.

3. Сложить полученные спектры со спектрами в GAM2 по программному модулю SUMME.

4. Выделить из суммарных спектров каналы 50-100 и собрать в специальном массиве по программному модулю SELECT .

5. Добавить к полученному массиву набор параметров PEAK.

6. Исключить PEAK, EXP1 и EXP2 из постоянного каталога.

7. Занести результаты задания в постоянный файл GAM1 , уничтожая старое содержимое.

Задание:

```
*CALL COC
      PROGRAM TEST1
      FILE GAM1, PEAK (UPDATE,TAPE01)
      FILE GAMMA (WORK)
      FILE GAM3 (WORK,1024)
      CREATE GAM2 (INPUT,TAPE01) = ABC,AA0001,AA0002
      DO 1 J = 1,3
      CALL CMULT (GAM1(J), 0.334, GAMMA(J))
      CALL SUMME (GAMMA(J),GAM2(J),GAMMA(J))
    1 CALL SELECT (GAMMA(J), 50, 100,GAM3)
      ADD PEAK TO GAM3
      WRITE GAMMA, GAM3 INTO GAM1 (1)
      DELETE PEAK, EXP1,EXP2(TAPE01)
      END
```

Задание является частью пакета, составленного по требованиям мониторной системы ЭВМ БЭСМ-6 /4/. Вызов системы COC происходит с помощью управляющей карты *CALL COC . Сформулированное на языке COC задание всегда начинается с оператора PROGRAM и имени программы. После этого описываются используемые файлы: постоянные файлы GAM1 и PEAK , рабочий файл GAMMA и бесструктурный рабочий файл GAM3 . Раздел описания кончается организацией нового постоянного файла GAM2 , который объединяет постоянные файлы ABC, AA0001 и AA0002. Программный модуль CMULT читает спектр из постоянного файла GAM1 и пишет результат в рабочий файл GAMMA . Модуль SUMME складывает спектры, которые находятся в файлах GAMMA и GAM2 . Результат пишется в тот же подфайл файла GAMMA , в который модуль CMULT записал свой результат. При этом предыдущая запись уничтожается. Модуль SELECT выделяет из спектров, находящихся в файле GAMMA , каналы 50-100

и переписывает их в бесструктурный рабочий файл GAM3 . После выполнения цикла постоянный файл PEAK перепишется в GAM3 . Запись в GAM3 происходит в режиме продолжения. Оператор WRITE переписывает рабочие файлы GAMMA и GAM3 в постоянный файл GAM1 . При этом уничтожаются старые подфайлы с GAM1(1) по GAM1(4). В конце задания исключаются файлы PEAK, EXP1 и EXP2 из постоянного каталога. Файлы EXP1 и EXP2 были ранее зарегистрированы в постоянном каталоге.

ЛИТЕРАТУРА

I. Л.С.Нефедьева. Совещание по программированию и математическим методам решения физических задач. Стр. 476-480, ОИЯИ, D10-7707, Дубна, 1974.

2. Х.Кэниг, Л.С.Нефедьева. Совещание по программированию и математическим методам решения физических задач. Стр. 514-517. ОИЯИ, D10-7707, Дубна, 1974.

3. Д.Лефковиц . Структуры информационных массивов оперативных систем. Пер. с англ. под редакцией О.И.Авена. Издательство "Энергия", Москва, 1974,

4. Г.Л.Мазный. Мониторная система "Дубна". ОИЯИ, 11-5974, Дубна, 1971.

Рукопись поступила в издательский отдел
27 января 1975 г.