



сообщения  
объединенного  
института  
ядерных  
исследований  
дубна

9  
4

1401/2-81

P10-80-860

Г.Л.Бутцева, Л.С.Нефедьева

РЕАЛИЗАЦИЯ НА ЯЗЫКЕ ФОРТРАН  
ДИРЕКТИВ СИСТЕМЫ  
ОБРАБОТКИ СПЕКТРОВ (СОС)

1980

Задание на обработку набора данных в системе СОС/2/ представляет собой последовательность директив<sup>1/</sup> на языке этой системы. Каждой директиве соответствует на ФОРТРАНЕ<sup>3/</sup> один или несколько операторов CALL, а также один или несколько декларативных операторов.

#### 1. Заголовок и признак конца записи задания

Задание начинается с фортранного оператора "PROGRAM имя" и кончается оператором "END".

В начале задания необходимо перечислить следующие стандартные операторы:

```
COMMON/CSFIL/FILE  
EXTERNAL CLDAD1,CLDADC  
CALL CPAGE1  
CALL CFWORK(FILE,1)  
CALL CBEGIN .
```

Данный набор операторов - эквивалент директивы СОС PROGRAM.

Перед фортранным оператором END, т.е. в конце задания, надо поставить перфокарту CALL SEND.

Пользователь может работать в двух режимах: в режиме большого буфера - 5К:

```
EXTERNAL CLDAD1 и CALL CPAGE1,
```

и в режиме малого буфера - 1К:

```
EXTERNAL CLDAD2 и CALL CPAGE2 .
```

Буфер служит временным хранилищем при выполнении операций обмена между файлами и записями. Использование большого буфера ускоряет указанные операции обмена. Однако в этом случае для программных модулей остается меньше места в оперативной памяти.

Если пользователь в задании использует рабочие ленты (ленты с записанными на них постоянными файлами), то в специально выделенную им ячейку заносится порядковый номер рабочей МЛ в виде восьмеричной константы.

Например, DATA(TAPE=1B) .

Перед оператором SWESIN пользователь должен разослать имена рабочих МЛ (они совпадают с именами МЛ в картах запроса ресурсов) в специальные системные таблицы с помощью программы STAPE . При обращении к этой программе в качестве фактических параметров задаются имена лент в текстовом виде. Если данная МЛ отсутствует, то задается 0. Подпрограмма STAPE имеет три параметра.

Пример обращения:

```
CALL STAPE('NATAL1',0,0).
```

Количество МЛ в настоящее время равно 2.

После описания всех файлов, используемых в задании, нужно поставить перфокарту

```
CALL SPAT0D.
```

При работе только с рабочими файлами обращение к системной программе STAPE сохраняется, но оно имеет вид

```
CALL STAPE(0,0,0).
```

Обращение к системной программе SPAT0D в этом случае не нужно.

## 2. Декларативные операторы

Всем декларативным операторам языка СОС соответствуют системные программы, обращение к которым происходит через оператор CALL, кроме

```
COMMON/CS1/ .
```

Под каждый файл, используемый в задании, в основной программе нужно выделить массив имени файла из двух ячеек. В первую ячейку оператором DATA заносится имя файла в виде текстовой

константы. Во вторую ячейку файла заносится номер записи, с которым пользователь хотел бы работать в данный момент.

### 2.1. Оператор описания файлов

Данный оператор предназначен для описания файлов, используемых в задании и зарегистрированных в системе до его выполнения. В зависимости от типа описываемых файлов обращение к программам, эквивалентных оператору FILE, может быть представлено в одной из следующих форм:

2.1.1. CALL CINPUT(FILE1, FILE2, ..., FILEN, N, TAPE),

где FILE1, ..., FILEN - имена файлов;  
N - число файлов;  
TAPE - порядковый номер МЛ (восьмеричная константа).

Файлы типа INPUT используются, если они уже зарегистрированы в паспорте МЛ.

Для файлов типа INPUT допускаются следующие операции обмена:

- 1) чтение;
- 2) запись в режиме расширения, т.е. добавления новых записей к уже имеющимся в данном файле.

При записи в файл номера записей следует указывать последовательно. В противном случае система прерывает задание и печатает сообщение "Номер записи неверен". При чтении несуществующей записи система работает с последней записью файла.

2.1.2. CALL COUTPUT(FILE1, ..., FILEN, N, TAPE).

Для файлов типа OUTPUT допускается запись в режиме расширения. Необходимо правильно указывать номера записей. Ошибочное задание система не контролирует и срабатывает неверно. Читать файлы типа OUTPUT нельзя. Система прерывает задание.

2.1.3. CALL CUPDAT(FILE1, ..., FILEN, N, TAPE).

Файлы типа UPDATE используются, если они зарегистрированы в паспорте МЛ. Для файлов типа UPDATE допускаются следующие операции обмена:

- 1) чтение;
- 2) запись в режиме расширения;
- 3) запись в режиме затирания.

Если в задании не указаны номера записей, система работает с последней записью файла.

## 2.2. Операторы создания новых файлов

2.2.1. Создание новых постоянных файлов осуществляется оператором `CALL CREAT(0, FILE, TAPE)`, где `FILE` - имя вновь создаваемого файла. Нельзя читать из вновь созданного файла. Система уходит по сбою на конец работы и печатает фразу "Нельзя читать, файл типа `OUTPUT`".

2.2.2. Создание новых постоянных файлов путем объединения существующих постоянных файлов. Оператор объединяет несколько постоянных файлов и их записи в один файл под новым именем. Имеются две возможности объединения: постоянное (типы файлов `INPUT` и `UPDATE`) и временное (тип файла `TIME`).

Примеры обращений:

А. `CALL CREATI(FILEN, ..., RECK, FILEK, ..., FILE1, ..., L, FILE, TAPE)`,

где `FILEN, ..., FILE1, ...` - имена файлов;  
`RECK, FILEK, ...` - их записи, из которых создается новый файл;  
`L` - число параметров (число файлов и их записей, указанных в списке);  
`FILE` - новый файл типа `TIME`;

Б. `CALL CREATU(FILEN, ..., RECK, FILEK, ..., FILE1, ..., L, FILE, TAPE)`,  
где `FILE` - новый файл типа `UPDATE`.

Если номер записи, указанный в списке, не существует, система будет работать с последней записью файла;

В. `CALL CREAT(FILEN, ..., RECK, FILEK, ..., FILE1, ..., L, FILE)`, где  
`FILE` - новый файл типа `TIME`.

Тип `TIME` соответствует типу `UPDATE`, но в паспорте МЛ не регистрируется и существует только в пределах одного задания. Режим перезаписи и расширения для файлов типа `TIME` недопустим. Обращение к этому оператору возможно и с указанием МЛ;

Г. `CALL CREATT(FILEN, ..., RECK, FILEK, ..., FILE1, ..., L, FILE, TAPE)`,  
где `TAPE` - порядковый номер МЛ.

### 2.3. Создание рабочих файлов

Файлы, которые создаются только на период выполнения данного задания, называются рабочими. Рабочим файлам присваивается признак `WORK`, который по своему действию соответствует признаку `UPDATE` для постоянных файлов.

#### 2.3.1. Рабочие файлы с произвольным доступом:

`CALL CFWORK(FILE1, ..., FILEN, N)`, где

`FILE1, ..., FILEN`

- имена рабочих файлов в данном задании;

`N`

- число рабочих файлов в этом операторе.

Для хранения рабочих файлов имеется память 50К слов на магнитных барабанах. Длина записи рабочего файла не должна превышать 4096 слов в I режиме и не ограничена (в разумных пределах) во II режиме. При записи в файл, если не указан номер записи, система записывает в 1-ю запись. При считывании номер записи нужно указывать обязательно.

#### 2.3.2. Рабочие файлы с последовательным доступом:

`CALL CFWORK(FILE1, ..., FILEN, N, LENGTH)`

, где

`LENGTH`

- длина рабочего файла с последовательным доступом.

Рабочие файлы с последовательным доступом служат для сборки отдельных числовых данных или наборов данных.

Максимальная длина файла не должна превышать 15К в режиме малого буфера и 4К в режиме большого буфера. Доступ к данным возможен только последовательным способом, для этого каждый файл имеет указатель. Программы `CREAD` и `CWRITE` читают или записывают данные с того места файла, на которое показывает указатель файла. Для управления указателем файла имеются следующие программы:

`CRESET(FILE)` - устанавливает указатель файла `FILE` на начало;

`CREWRT(FILE)` - устанавливает указатель файла `FILE` на начало и уничтожает содержимое файла;

- SWACK (FILE) - устанавливает указатель файла FILE на предыдущую запись данных;
- STBEFF (FILE) - устанавливает указатель файла на настоящий конец файла;
- SEBF (FILE) - является FUNCTION, позволяющей определить положение указателя файла.

Для чтения данных из файла этого типа используется программа CREAD. Обращение:

CALL CREAD (FILE, NUMBER, FIELD, FFORMAT).

Запись данных реализуется оператором

CALL CWRITE (FIELD, NUMBER, FILE, FFORMAT),

- где FILE - имя файла, из которого читаются или в который записываются данные;
- NUMBER - число записей (число данных);
- FIELD - имя массива, в который или из которого передаются данные, FIELD может быть простой переменной, если NUMBER=1;
- FFORMAT - признак формата (указывается числом).

#### 2.4. Оператор описания общего блока COMMON

Оператор COMMON записывается по правилам языка ФОРТРАН в следующей форме:

COMMON /сѳс1/ <идентификатор> (длина).

Массивы, которые описываются в задании, можно расположить в общем COMMON-блоке сѳс1. Его размер определяется пользователем, минимальный размер 4200 слов. Запрещается передача данных между модулями через COMMON-блок сѳс1, т.к. некоторые системные операторы используют его. Он вводится с целью экономии памяти машины.

### 3. Выполняемые операторы

К выполняемым операторам относятся

WRITE, RENAME, DELETE, CALL, Dѳ.

Операторы CALL и Dѳ используются по правилам языка ФОРТРАН.

### 3.1. Оператор обмена записями

Обмен записями осуществляется оператором

```
CALL SWRIT (RECN, FILEN, ..., REC1, FILE1, N, RECC, FILE) ,
```

- где
- |                   |   |
|-------------------|---|
| FILE1, ..., FILEN | - имена файлов и их записи, откуда переписывается информация. Если номер записи равен 0, то передается весь файл; если указан номер записи, то выбирается одна запись;  |
| REC1, ..., REC N  |   |
| N                 | - число параметров (число файлов и их записей);   |
| RECC, FILE        | - запись и файл, куда записывается информация. Если номер записи равен 0, обмен идет в режиме продолжения; если указан конкретный номер записи, то информация записывается именно с этого места и затирается старая информация. |

Допускаются следующие виды обмена:

постоянный файл  $\longleftrightarrow$  постоянный файл,  
рабочий файл  $\longleftrightarrow$  рабочий файл,  
постоянный файл  $\longleftrightarrow$  рабочий файл.

Все файлы, между которыми идет обмен, нужно описывать, иначе система прерывает задание и печатает сообщение "Тип файла не описан". Если неправильно указана запись, начиная с которой нужно вести обмен, система печатает сбойное сообщение "Номер записи неверен".

### 3.2. Оператор переименования файлов

А. Обращение без указания M1

```
CALL CRENAM (NEWN, OLDN, ..., NEW1, OLD1, L) ,
```

- где OLDN - имя файла, подлежащее переименованию;  
NEWN - его новое имя;  
L - число параметров.

Б. Обращение с указанием M1

```
CALL CRENAT (NEWS, OLDS, ..., NEW1, OLD1, L, TAPE) .
```

Пример:

```
CALL CRENAM (C, A, D, B, 4) .
```



При срабатывании этого оператора переменится имя файла А на С, а имя файла В на D. Имена файлов (старые) в паспорте затираются. Можно переименовывать файлы независимо от типа и вида, но все эти файлы должны быть описаны. Максимально возможное число файлов и записей при работе с этим оператором - 15. В одном задании возможно обратное переименование новых файлов в старые.

### 3.3. Оператор переименования записей файла в самостоятельные файлы

А. Обращение с указанием МЛ:

```
CALL CNEWFT(NEWN,...,NEW1,OLDFIL,L,TAPE).
```

Б. Обращение без указания МЛ:

```
CALL CNEWFI(NEWN,...,NEW1,OLDFIL,L),
```

где `OLDFIL` - имя файла, подлежащее переименованию;  
`NEWN,...,NEW1` - новые имена его записей. Соответствие в обратном порядке;  
`L` - число новых самостоятельных файлов.

Пример:

```
CALL CNEWFI(A10,...,A1,A,10).
```

При выполнении этого оператора 10 записей файла А переименуются в самостоятельные файлы, состоящие из одной записи: А1,... А10. В паспорте МЛ информация о старом файле А затирается.

Перед обращением к оператору переименования записей исходного файла в самостоятельные файлы исходный файл должен быть описан. В случае работы с несуществующей записью файла система работает с последней записью этого файла.

### 3.4. Оператор исключения файлов и записей

А. Обращение с указанием МЛ:

```
CALL CDELET(FILEN,...,RECK,FILEK,...,FILE1,K,TAPE).
```

Б. Обращение без указания МЛ:

```
CALL CDELE(FILEN,...,RECK,FILEK,...,FILE1,K),
```

где `FILEN,...,FILE1` } - имена файлов и их записи, которые подлежат исключению;  
`RECK,FILEK` }  
`K` - число параметров.

Пример:

```
CALL CDELE(10,B,A,3).
```

После выполнения этого оператора будет исключен файл А, в файле В произойдет исключение 10-й записи и перенумерация оставшихся записей.

Данный оператор можно использовать только для записей постоянных файлов. Список файлов и записей не должен превышать 15 элементов. Все исключаемые файлы должны быть описаны. Нельзя исключать в одном задании вновь созданные файлы. Система прерывает задание и печатает сообщение "Нельзя исключать вновь созданный файл". В случае задания несуществующей записи система работает с последней записью файла.

#### 4. Сегментация пакета задания на языке СОС

При нехватке оперативной памяти рекомендуется следующая последовательность действий:

1. Увеличить память на 8К с помощью управляющей карты <sup>/4/</sup>  
`*CALL PISMEMORY .`

2. Если данное задание работало с большим буфером - перейти на малый буфер, т.е. заменить п/к `EXTERNAL CLØAD1` и `CALL SPAGE1` на `EXTERNAL CLØAD2` и `CALL SPAGE2`. Это сэкономит 4К памяти.

3. Организовать сегментацию задачи на отдельные разделы, сменяющие друг друга в памяти во время счета. Тогда задание пользователя рекомендуется оформлять следующим образом: за головной программой следует сегмент, в который внесены все операторы описания файлов, произведена загрузка внутренних `COMMON`-блоков и т.д.

Следующий сегмент составляет задание пользователя. Пользователь из этого сегмента может обращаться к другим программным модулям, может производить любые операции над файлами (в рамках системы СОС).

#### Замечания

1. Обращения к любому программному модулю может быть организовано через оператор `CALL LØADGØ`.

2. Программные модули из библиотеки СОС нельзя разбивать на сегменты.

### 5. Пример

На МЛ с именем SISTY8 имеется файл с именем TNS. Его надо переписать в рабочий файл TON; обработать программным модулем SPEKTR участок спектра с 5 по 20 канал. Результат записать на эту же МЛ. Распечатать результат обработки в виде таблицы. Распечатать паспорт МЛ:

```
PROGRAM PRIM1
COMMON/COC1/A(4200)
EXTERNAL CLOAD2
COMMON/COCFIL/FILE(2)
EXTERNAL CLOADC
INTEGER L1(2),L2(2),L3(2)
DATA(TAPE=1B)
DATA(L1(1)=3HINS), (L3(1)=3HTON), (L2(1)=4HLIDA)
CALL CPAGE2
CALL CFWORK(FILE,1)
CALL CTAPE('SISTY8',0,0)
CALL CBEGIN
CALL CINPUT(L1,1,TAPE)
CALL CREATO(0,L2,TAPE)
CALL CFWORK(L3,1)
CALL CPATOD
L1(2)=1
L2(2)=1
L3(2)=1
CALL CWRIT(1,L1,2,0,L3)
CALL SPEKTR(L3,5,20,L2)
CALL CPRINT(L2,5,20)
CALL CPASSP(TAPE,2)
CALL CEND
STOP
END
```

Этот же пример, но в режиме сегментации:

```
PROGRAM PRIM2
EXTERNAL CLOAD2
INTEGER L1(2),L2(2),L3(2)
DATA(TAPE=1B)
DATA(L1(1)=3HINS),(L3(1)=3HTON),(L2(1)=4HLIDA)
CALL CPAGE2
CALL LOADGO(L1,L2,L3,TAPE,'CDECL')
CALL LOADGO(L1,L2,L3,TAPE,'CEXEC')
STOP
END
```

```
SUBROUTINE CDECL(L1,L2,L3,TAPE)
EXTERNAL CLOADC
COMMON/COCFIL/FILE
CALL CFWORK(FILE,1)
CALL CTAPE('SISTY8'O,O)
CALL CBEGIN
CALL CINPUT(L1,1,TAPE)
CALL CREATO(O,L2,TAPE)
CALL CFWORK(L3,1)
CALL CPATOD
RETURN
END
```

```
SUBROUTINE CEXEC(L1,L2,L3,TAPE)
INTEGER L1(2),L2(2),L3(2)
COMMON/COC1/A(4200)
L1(2)=1
L2(2)=1
L3(2)=1
CALL CWRT(1,L1,2,O,L3)
CALL SPEKTR(L3,5,20,L2)
CALL CPRINT(L2,5,20)
CALL CPASSP(TAPE,2)
CALL CEND
RETURN
END
```

Литература

1. Кэниг Х., Нефедьева Л.С., Салтыков А.И. Язык общения в автоматизированной системе обработки физической информации. Москва, "Приборпром", 1975.
2. Кэниг Х. и др. Автоматизированная система обработки спектров (СОС) на машине БЭСМ-6. Рига, "Зинатне", 1975.
3. Язык ФОРТРАН/под ред. В.П.Шурикова/.ОИЯИ, ПП-4818, Дубна, 1969.
4. Мазный Г.Л. Программирование на БЭСМ-6 в системе "Дубна". "Наука", М., 1978.

Рукопись поступила в издательский отдел  
26 декабря 1980 года.