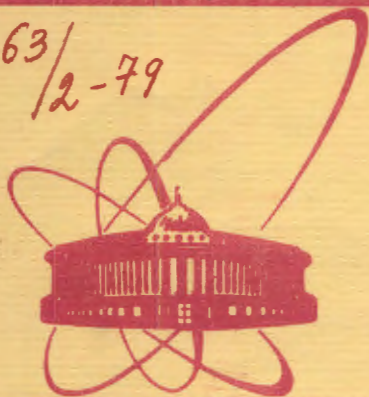


2263/2-79



СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА

Ц 8405  
Д-691

11/vi-79  
10 - 12277

В.И.Дорогов, М.Кунике

ПОДПРОГРАММЫ ДЛЯ ДОСТУПА  
К НЕКОТОРЫМ СИСТЕМНЫМ КОМПОНЕНТАМ  
ОС ЕС ИЗ ЯЗЫКА ФОРТРАН (I)

1979

10 - 12277

В.И.Дорогов, М.Кунике

ПОДПРОГРАММЫ ДЛЯ ДОСТУПА  
К НЕКОТОРЫМ СИСТЕМНЫМ КОМПОНЕНТАМ  
ОС ЕС ИЗ ЯЗЫКА ФОРТРАН (I)

Объединенный институт  
ядерных исследований  
БИБЛИОТЕКА

Дорогов В.И., Кунике М.

10 - 12277

Подпрограммы для доступа к некоторым системным компонентам ОС ЕС из языка ФОРТРАН (I)

Приводятся описание и примеры использования фортрановских подпрограмм DIALOG и COMMND для внесения информации с консоли по инициативе подпрограммы и оператора, а также описание подпрограмм для вызова процедур на языке PL/I из программы на языке ФОРТРАН. Рассматривается вопрос передачи данных из процедур программы на языке PL/I в подпрограмму на языке ФОРТРАН.

Подпрограммы написаны на языке АССЕМБЛЕР и работают под управлением операционной системы ОС ЕС любой версии.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1979

Dorogov V.I., Kunicke M.

10 - 12277

Subroutines for Access to Several OS/EC Components from FORTRAN Language. I

Description and examples of FORTRAN subroutines DIALOG and COMMND for interaction with the operator console and PL/I subroutine for call of the PL/I language procedures from FORTRAN language are given. Subroutines are written in ASSEMBLER language and are applied to the OS/EC operating system.

The investigation has been performed at the Laboratory of High Energies, JINR;

Communication of the Joint Institute for Nuclear Research. Dubna 1979

Язык ФОРТРАН-IV не дает возможности внесения информации с консоли в программу во время ее выполнения. Эти функции можно запросить из программы на языке АССЕМБЛЕР, используя системные компоненты операционной системы ОС ЕС. Подпрограммы DIALOG и COMMND предназначены для обращения к системным компонентам ОС ЕС: запроса текущей программы и выдачи ее ответа при помощи макрокоманды WTOR, а также операторских команд STOP и MODIFY.

Другой проблемой, решаемой в работе, является осуществление возможности вызова процедур на языке PL1 из программы на языке ФОРТРАН-IV через подпрограмму PL1 и установления соответствия между данными этих языков.

Для всех подпрограмм приведены примеры обращения. При этом предполагается, что вызываемые подпрограммы записаны в библиотеке загрузочных модулей USR1.PGM LIB.

#### ПОДПРОГРАММА ЗАПРОСА ИНФОРМАЦИИ С ОПЕРАТОРСКОЙ КОНСОЛИ DIALOG

В ряде случаев по запросу программы, написанной на языке ФОРТРАН-IV, необходимо передать информацию, например для продолжения работы, модификации данных, изменения ветви и т.п. Если количество этой информации не слишком большое, удобно использовать для этих целей консоль оператора.

Подпрограмма DIALOG предназначена для выдачи запроса на консоль оператора и передачи информации с консоли в фортрановскую программу.

## Структура подпрограммы DIALOG

Подпрограмма

имя подпрограммы: DIALOG

вызываемых программ нет

вызов: CALL DIALOG (< сообщение >, < адрес ответа >, < длина ответа >)

язык программирования: АССЕМБЛЕР

операционная система: ОС/ЕС

## Использование подпрограммы DIALOG

DIALOG вызывается как подпрограмма фортрановским оператором CALL, написанным в соответствующих местах фортрановской программы.

Параметры программы. Сообщение задает либо адрес памяти (идентификатор массива или переменной), по которому размещается сообщение, которое появится на консоли оператора, либо непосредственно сам текст сообщения в виде строчной константы. Длина сообщения фиксирована и равна 48 символам. Адрес ответа определяет адрес памяти (имя массива или переменной), по которому будет занесена информация с консоли. Длина ответа задается переменной типа INTEGER\*4 или числовой константой и определяет максимальную длину ответа в символах, значение которой не должно превышать 118.

Алгоритм программы. Подпрограмма построена на использовании макрокоманды супервизора WTOR. При каждом обращении к подпрограмме происходит выдача сообщения на консоль оператора. После этого выполнение программы прерывается до поступления ответа. Набранная на консоли информация заносится по указанному в параметре обращения адресу.

Пример вызова подпрограммы DIALOG

```
// EXAMPLE      JOB      HUSER, NAME
//              EXEC     FORSTCLD
// FORT.SYSIN   DD        *
```

```
REAL*8 REPLY (2), TEXT (6)
DATA TEXT /'I WISH Y', 'OU A NAP', 'PY NEW Y', 'EAR'/
CALL DIALOG (TEXT, REPLY, 16)
PRINT 100, REPLY
CALL DIALOG ('THE PARAMETER IS A HOLLERITH
* CONSTANT OF 48 CHAR', REPLY, 16)
PRINT 100, REPLY
STOP
100 FORMAT (3X, 2A8)
END
/*
// GO.SYSLIB DD
//          DD DSN=USR1.PGMLIB,DISP=SHR
//
```

На операторскую консоль при выполнении этой программы выдаются следующие сообщения, требующие ответа оператора:

```
*nn USR000A I WISH YOU A HAPPY NEW YEAR      Pmm
```

и

```
*nn USR000A THE PARAMETER IS A HOLLERITH CONSTANT OF
48 CHAR Pmm
```

## ПОДПРОГРАММА-ФУНКЦИЯ ДЛЯ ИЗМЕНЕНИЯ ХАРАКТЕРИСТИК И ОСТАНОВКИ ПРОГРАММЫ COMMND

При выполнении фортрановской программы иногда возникает необходимость изменить ход выполнения программы с операторской консоли.

Примером такого случая может служить остановка задания с консоли с помощью команды STOP, а не команды CANCEL, использование которой приводит к аварийному завершению задания.

Подпрограмма-функция COMMND дает возможность применить операторские команды MODIFY и STOP для данного задания.

## Структура программы COMMND

Подпрограмма-функция

имя программы: COMMND

вызываемых подпрограмм нет

вызов: COMMND ( < область параметров > )

язык программирования: АССЕМБЛЕР

операционная система: ОС/ЕС

## Использование программы COMMND

К COMMND следует обратиться как к подпрограмме-функции в соответствии с правилами языка ФОРТРАН- IV. Значение функции указывает наличие или отсутствие операторской команды MODIFY или STOP для данного задания. Если COMMND=0, то не было команды, при COMMND=+1. была задана команда STOP, а при COMMND=-1.- команда MODIFY. Для определения наличия команды необходимо в ходе выполнения программы периодически обращаться к функции COMMND.

Параметр программы. Область параметров определяет адрес памяти (идентификатор массива или переменной), по которому заносится информация, набранная на консоли, в команде MODIFY. Адрес памяти должен быть на границе полуслова, т.е. тип массива или переменной не должен быть LOGICAL \*1. В первом полуслове области параметров записана длина информации из команды MODIFY.

Алгоритм работы программы. Подпрограмма-функция использует системные макросы EXTRACT и QEDIT для организации работы с буфером ввода команд (CIB). При первом в данной программе обращении к COMMND разрешается построение добавочного буфера ввода команд для данного задания. При каждом следующем обращении COMMND просматривает буфер ввода команд и в соответствии с результатом просмотра устанавливает значение функции. В случае наличия команды буфер освобождается и разрешается построение нового. При наличии команды MODIFY текст из операторской команды передается в вызывающую фортрановскую программу.

## Пример вызова подпрограммы-функции COMMND

```
// EXAMPLE      JOB   HUSER, NAME
//              EXEC  FORSTCLD
// FORT.SYSIN   DD   *
                INTEGER*2 TEXT(21)
                INTEGER*2 COUNT
                LOGICAL*1 TEXTBT (40)
                EQUIVALENCE (TEXT(1), COUNT)
                EQUIVALENCE (TEXT(2), TEXTBT(1))
400 IF (COMMND(TEXT)) 100, 200, 300
100 PRINT 110, (TEXTBT (I), I=1, COUNT)
200 CONTINUE
.
.
.
GOTO 400
300 CONTINUE
.
.
.
STOP
END

/*
// GO.SYSLIB   DD
//              DD DSN=USR1.PGMLIB, DISP=SHR
//
```

Команда MODIFY может выглядеть, например, так:  
M EXAMPLE, 'I WISH TO MODIFY PARAMETERS'  
В данном случае текст команды будет выдан на устройство системного вывода.

Команда STOP для данного задания выглядит следующим образом:

#### Р EXAMPLE

#### ОРГАНИЗАЦИЯ СВЯЗИ FORTRAN - PL/1

При решении задачи совместного использования языков FORTRAN и PL/1 в одной программе возникают некоторые ограничения, обусловленные в основном отличиями в способе, которым обрабатываются данные в языке PL/1, в частности способами планирования и запоминания (т.е. средой языка PL/1), и большим количеством типов данных, имеющихся в языке PL/1.

#### PL1 - ФОРТРАНОВСКАЯ ПОДПРОГРАММА ДЛЯ ВЫЗОВА ПРОЦЕДУР PL/1

Основной проблемой несовместимости языков FORTRAN и PL/1 является различие сред этих языков. Эта проблема решается созданием секции сопряжения, написанной на языке АССЕМБЛЕР, для вызова процедур на языке PL/1 из программы на языке ФОРТРАН.

#### Структура подпрограммы PL1

Подпрограмма

имя программы: PL1

вызываемых программ нет

вызов: CALL PL1 (< имя >, < параметр 1 > ,  
< параметр 2 > , ..., < параметр N > )

язык программирования: АССЕМБЛЕР

операционная система ОС/ЕС

#### Использование подпрограммы PL1

Подпрограмма PL1 вызывается из языка ФОРТРАН оператором CALL в любом месте программы.

Параметры подпрограммы. Параметр < имя > задает имя процедуры на языке PL/1, которая вызывается при данном обращении из языка ФОРТРАН. Имя этой процедуры необходимо описать как EXTERNAL на языке ФОРТРАН. < параметр 1 > , ..., < параметр N > - фактические аргументы для процедуры на языке PL/1.

Алгоритм работы подпрограммы. Основная цель секции сопряжения состоит в том, чтобы сохранить среду языка, вызвавшего подпрограмму, и установить и сохранить для следующих вызовов среду языка, которому она передает управление.

Подпрограмма PL1 выполняет следующие функции:

а) при первом вызове секции сопряжения устанавливается среда PL/1 при помощи вызова библиотечного модуля IHESAP в точке входа IHESAPD и выполняются следующие действия:

- иницируются псевдореестры и рабочая область библиотеки LWS;

- находится длина вектора псевдореестров (PRV) и PRV размещается в динамической области памяти данных (PRV VDA);

- далее, при первом и при всех дальнейших вызовах иницируется процедура на языке PL/1, т.е. отключается фортрановский модуль обработки ошибок и устанавливается модуль обработки ошибок IHERR для языка PL/1; PRV VDA сцепляется с внешней областью сохранения;

б) иницируется DSA и соединяется с другими DSA стандартным для языка PL/1 образом;

в) вызывается точка входа процедуры на языке PL/1, производится передача параметров;

г) после возврата управления в секцию сопряжения сохраняются указатель PRV и адрес модуля обработки ошибок для языка PL/1;

д) осуществляется возврат в подпрограмму, вызывающую процедуру языка PL/1, и восстанавливается модуль обработки ошибок для языка ФОРТРАН.

## Замечания

1. Процедуры на языке PL/1 должны быть неглавными внешними процедурами.

2. На шаге GO необходимо соединить библиотеки SYS1.PL1LIB и SYS1.FORTLIB.

## ОБМЕН ДАННЫМИ МЕЖДУ ПРОГРАММИСТАМИ НА ЯЗЫКАХ ФОРТРАН И PL/1

Основные области несовместимости, связанные с различием в представлении данных на языках ФОРТРАН и PL/1, следующие:

- язык PL/1 имеет больше типов данных, поэтому некоторые из них, а именно, литернострочные данные, не имеют эквивалента на языке ФОРТРАН;
- записи языка PL/1 могут быть организованы как структуры в основной памяти (с выровненными и невыровненными элементами). Записи языка ФОРТРАН строятся при выводе и элементы не будут выравниваться, если только они не записаны в специальном порядке;
- массивы языка PL/1 запоминаются в порядке старшинства строки, в то время как массивы языка ФОРТРАН запоминаются в порядке старшинства колонки.

## ЭКВИВАЛЕНТНЫЕ ТИПЫ ДАННЫХ

В табл. 1 показаны эквивалентные типы данных для языков ФОРТРАН и PL/1.

## Структуры и выравнивание

Когда структуры языка PL/1 связываются с переменными языка ФОРТРАН, самый надежный путь - объявить структуры UNALIGNED. Однако если элементы записи языка ФОРТРАН были правильно выровнены (с использованием таких средств, как запись переменных в порядке понижения длины данных), структура на языке PL/1 может быть объявлена ALIGNED.

Таблица

ФОРТРАН		PL/1		Выравнивание	
Тип данных	Длина в байтах	Тип данных	Длина в байтах	Выровн.	Невыровн.
INTEGER*2	2	FIXED BINARY(15)	2	полуслово	байт
INTEGER*4	4	FIXED BINARY(31)	4	полное слово	"--
REAL*4	4	REAL FLOAT(корот.)		слово	"--
REAL*8	8	REAL FLOAT(длинн.)	8	двойное слово	"--
COMPLEX*8	8	COMPLEX FLOAT	8	полное слово	байт
COMPLEX*16	16	COMPLEX FLOAT	16	двойное слово	"--
LOGICAL*1	1	BIT(8)	1	байт	"--
LOGICAL*4	4	BIT(32)	4	байт	"--
		CHARACTER	любая	байт	"--

Массивы. Для многомерных массивов решение проблемы состоит в том, чтобы описать на одном языке массив с индексами в порядке, обратном по отношению к порядку описания на другом языке, например:

FORTRAN

REAL A(2,5,7)

PL/1

DCL A(7,5,2)

Другой способ - использование LSUB -определения в программе на языке PL/1, например:

FORTRAN

REAL A(7,5,2)

PL/1

DCL A1(7,5,2),

A(2,5,7) DEF A1(3SUB, 2SUB, 1SUB);

Теперь, если запись передается в A1 или из него, обращение к конкретному элементу в программе на языке ФОРТРАН будет идентично обращению к этому элементу на языке PL/1.

Внешние представления чисел

Несовместимость языков PL/1 и ФОРТРАН во внешнем представлении чисел с плавающей точкой в потокоориентированных наборах данных выражается в следующем:

- для языка PL/1 показатель степени любого числа с плавающей точкой указывается литерой E. Для языка ФОРТРАН показатель степени числа с плавающей точкой "ординарной точности" указывается литерой E, "двойной точности" - литерой D;

- для языка PL/1 показатель степени числа с плавающей точкой, если он положителен, может дополнительно иметь знак "+". Если знака нет, показатель степени следует сразу за литерой E. Например:

0.01873E22 или 0.01873E+22.

Для языка ФОРТРАН, если показатель степени числа с плавающей точкой положителен, между литерой E или D и показателем степени может стоять пробел, а не знак "+". Например:

0.01873E 22

Эти отличия мешают прямому использованию средств потокоориентированной передачи для чтения или записи чисел с плавающей точкой, которыми можно обмениваться с программой на языке ФОРТРАН.

Передача элементов данных

При передаче элементов данных между модулями на языках PL/1 и ФОРТРАН возникает необходимость передачи литернострочных типов переменных и массивов, обращение к которым на языке PL/1 осуществляется через информационные вектора, не имеющие аналога в языке ФОРТРАН.

Один путь решения этой проблемы - произвести наложение базированных переменных в программе на языке PL/1 на начало элементов, которые запрашивают информационные векторы. Таким образом, можно сделать так, что параметры будут выглядеть как арифметические элементы в списке параметров.

Например, если строка должна передаваться модулю на языке PL/1 из модуля на языке ФОРТРАН:

```
PL1SUB: PROC (PARMA);
          DCL PARMA FIXED DEC (1, 0),
          STRINCA CHAR (20) BASED (P),
          P = ADDR (PARMA);
          .
          .
          .
```

Другой путь связи элементов данных между модулями языков ФОРТРАН и PL/1 - использовать поименованную общую область для элементов: COMMON на языке ФОРТРАН и STATIC EXTERNAL на языке PL/1, например:



## FORTRAN

```
COMMON /COMAR/ DVAR, I, A, CHR, IRAY  
DIMENSION DVAR (7), CHR(3), IRAY(100)
```

### PL/1

```
DCL 1 COMAR STATIC EXTERNAL,  
2 I FIXED BIN (31, 0),  
2 A FLOAT DEC,  
2 CHR CHAR (12),  
2 IRAY(100) FIXED BIN(31,0);
```

Пример вызова процедур языка PL/1  
на языке ФОРТРАН

```
// EXAMPLE      JOB      HUSER,NAME  
// FORTRAN      EXEC     FORTGC  
// FORT.SYSIN   DD       *  
      EXTERNAL  DAT1, DAT2  
      LOGICAL*1 DTE(8)  
      DIMENSION MASSIN(10)  
      CALL  PL1(DAT1,DTE)  
      .  
      .  
      CALL  PL1(DAT2, MASSIN, ISUMA, IDLINA)  
      STOP  
      END  
// PL1ST        EXEC     PL1LFCG  
// PL1.SYSIN    DD       *  
      DAT1: PROC (ZKPAR);  
      DCL  ZKPAR FIXED(1), ZK CHAR(8) BASED(PTR), ZW CHAR(8);  
      PTR = ADDR (ZKPAR);  
      ZK = DATE;  
      ZW=SUBSTR(ZK,5,2)'. 'SUBSTR(ZK,3,2)'. 'SUBSTR(ZK,1,2);  
      ZK=ZW;  
      PUT SKIP;
```

```
PUT LIST (ZK);  
END DAT1;
```

```
* PROCESS;
```

```
DAT2: PROC (FIELD, SUM, LENGTH);  
DCL FIELD FIXED(1), BASFIELD(1) BASED (PTR);  
PTR = ADDR (FIELD);  
SUM = 0;  
DO I=1 TO LENGTH;  
SUM = SUM + BASFIELD(I);  
END  DAT2;
```

```
// GO.SYSLIB  DD  
//           DD  DSN=USR1.PGMLIB, DISP=SHR  
//           DD  DSN=SYS1.FORTLIB, DISP=SHR  
//
```

Авторы считают своим приятным долгом выразить признательность системным программистам комбината "Роботрон" (г.Дрезден) за любезно оказанную помощь в изучении некоторых аспектов функционирования операционной системы ОС ЕС.

Рукопись поступила в издательский отдел  
1 марта 1979 года.