СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

E10 - 10543

V.P.Shirikov, D.C.Marinescu

THE PARAMETER PASSING MECHANISM IMPLEMENTATION IN DIFFERENT FORTRAN COMPILERS

1977

E10 - 10543

V.P.Shirikov, D.C.Marinescu*

THE PARAMETER PASSING MECHANISM IMPLEMENTATION IN DIFFERENT FORTRAN COMPILERS

* On leave of absence from the Institute for Atomic Physics, Bucharest, Romania

Ширикov В.П., Маринеску Д.К.

E10 - 10543

Применение и реализация механизма передачи параметров в различных трансляторах с языка ФОРТРАН

В работе содержится характеристика типов передачи параметров между различными программными модулями и способа реализации этих типов в различных вариантах трансляторов (в основном, трансляторов с языка ФОРТРАН). Изложение ведется применительно к используемым в настоящее время системам математического обеспечения ЭВМ серии ЕС, ЭВМ фирмы СДС и IBM , ЭВМ БЭСМ-6.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1977

Shirikov V.P., Marinescu D.C.

E10 - 10543

The Parameter Passing Mechanism Implementation in Different FORTRAN Compilers

This report contains some considerations, remarks and comparison for the types and methods of parameter transmissions between different programming modules and also for the methods of implementation for these transmissions in different compilers versions (first of all, FORTRAN compilers). The compilers under consideration are the compilers for the software systems which are used now with EC, CDC, IBM and BESM-6 computers.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1977

It is very important for any programmer to recognize in which way the parameters are communicated from one module to another for every implementation of the programming languages available.

For most programming languages this talk is not so different since the designers of the compilers have no other choice but to implement that particular type of call as stated by people who have designed the language. For example:

- in ALGOL the default argument transmission mechanism is by name; it can be changed to transmission by value by specifying the parameter as such in the procedure header.

- In PL/1 the default argument transmission mechanism is by reference; it changes to transmission by value when the argument is enclosed in an extra set of paranthesis, in the CALL statement or in the function reference. Also transmission by value arises when the argument is a constant or involves operators.

- In COBOL the arguments are transmitted by reference.

As far as FORTRAN is concerned several approaches are used:

- the strategy announced in CDC manuals [1,2] is said to be the call by name with the call by value of expressions appearing as actual parameters; also the call by value is used by internal functions. This approach applies to both FTN and RUN compilers available on CYBER machines.
- the IBM approach[3] used in FORTRAN F , G , H compilers makes use of the call by value result, parameter passing mechanism, as default, with the user option to request the call by reference, by enclosing the dummy arguments between slashes.

The practical implications of such different points of view are: sometimes the same program compiled by different FORTRAN com-

pilers (and even at different optimization levels of the same compiler) gives different results when executed.

This might look queer for someone not aware of the fact that for FORTRAN the convention concerning the parameter passing mechanism is lax and somehow it is left to the compiler designer to decide about the parameter passing mechanism.

To illustrate this idea we present in table 1 the results obtained when the main program and the subroutine from example # 1 were translated by several compilers available around.

```
      X = 1.0                          SUBROUTINE SUB (A,B,C,D)
      Y = 2.0                          B = A + A
      Z = 7.0                          D = A + C
      CALL SUB (X,X,X+Y,Z)             RETURN
      PRINT 1,X,Y,Z                    END
    1 FORMAT(1H, 3F15.1)
      STOP
      END
```

EXAMPLE 1.

COMPILER	TYPE OF CALL, ACCORDING TO THE MANUALS	VALUES X	Y	Z	OBSERVATIONS
FTN OPT=0	Call by name	2	2	5	User request OPT=0
FTN OPT=1	Call by name	2	2	4	User request OPT=1
FFORTRAN	Call by value/ result	2	2	4	Used the default type of call.
FFORTRAN	Call by reference	2	2	5	The user requested call by reference.
RUN	Call by name	2	2	4	
BESM-6	Call by reference	2	2	5	

TABLE 1.

Let us briefly examine the four basic types of parameter communication mechanisms and some details of their practical implementation.

- A./ Call by reference

It is eventually the easiest to implement and practically consists of passing the addresses of the actual arguments to the called program. As far as our example 1 is concerned, the expression:

W = X + Y

is first evaluated and a stack consisting of the addresses of X, X, W and Z is constructed (see appendix 1.2 and figure #1)

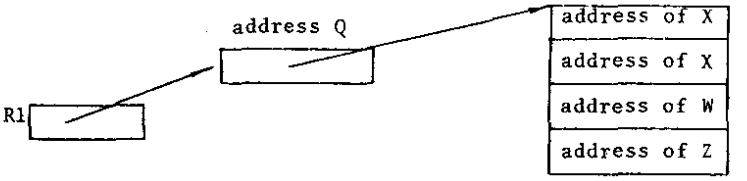

FIGURE #1. The call by reference implementation in FORTRAN F compiler.

The calling program loads in general purpose register R1 the address of a memory location Q where a pointer to the stack is to be found. It is the responsibility of the linkage editor to fill Q with the stack address.

When entered the subroutine copies the stack and throughout the computation operates with the locations in the main storage for X, X, W and Z, as defined by the calling program.

In this situation , side effects can occur; one of the most unpleasant ones occures when a constant is passed as an actual argument and the called module attempts to modify the dummy argument to which the constant is associated. In case of example #2 presented below, most FORTRAN compilers would lead to the result J=11.

```
      N = 8                          SUBROUTINE S (I,J,K)
      .....                          K = K + J
      M = 6                          I = I + 2
      J = 8                          RETURN
      CALL S (1,M,N)                 END
      J = J + 1
      PRINT 1,J
1     FORMAT (1H , I5)
      STOP
      END
```

EXAMPLE #2.

Sometimes to protect against unpleasant surprises, FORTRAN manuals [3] warn that "input parameters must not be changed inside the called module" (must not appear as the left-hand side of an assignation statement).

Another type of side effect is due to the fact that the call by reference is associated with the call by value of expressions appearing as actual parameters, as in the case of example #1; there, the expression is first evaluated using the old value for X. This type of side effect does not occur if the rule of not modifying the input parameters is observed and if every variable appearing in the expression is considered as an input parameter.

- B./ Call by value

Instead of passing the parameter addresses , in this case their values are communicated in a stack.

If the machine has enough registers the values of the parameters can be passed into the registers, thus saving time.

It should be noticed that there is no feedback from the called module to the calling one so that, though this strategy is most suitable for input parameter passing, it leaves no hope for output parameter communication. This is precisely the reason why this type of call has a limited use. As stated before the FTN compiler uses this type of call for internal functions since only input parameters are to be passed, their number is limited and anyway

the result is returned as a variable with the same name as the function name.

- C./ Call by value/result

As in the previous case the values of the actual parameters are passed but also a pointer to a stack of parameter addresses is communicated. As a result, the called module opperates with the values passed, but before returning to the calling module the values of actual arguments are updated.

If we examine appendix 1.2 we see that when the subroutine SUB is entered, the general purpose register R1 contains the address Q, where a pointer to the stack of addresses is inserted by the linkage editor; a copy of parameter values is made (the sequence of instructions starting at label A20). Then the computation suggested by statements 2 and 3 of the subroutine is performed (the sequence of instructions starting at label A52).

Before returning, R1 is loaded with the address of a memory location, where a pointer to the stack of parameter addresses is to be found (the sequence of two instructions starting at label A36), then all four parameters X,X,W,Z get their values updated (the four pairs of instructions, load and move constant).

It results that in addition to the side effects previously encountered, in connection with the call by reference, here occures an additional one; the computations performed by the called module are using the initial values of the input parameters.

- D./ Call by name

The call by name seems to be the most natural to communicate parameters since it is ment to implement the textural substitution and thus no side effects can possibly occur.

In case of our example #1, A is substituted by X, B by X, C by X+Y, D by Z and the executable statements of the subroutine are:

```
X = X+X ; /X = 1+1 = 2/
Z = X+(X+Y);/Z = 2+2+2 = 6/.
```

Obviously since the names of variables have a significance only within a module some sophisticated mechanism must be available so that when a reference is made to X in the called module, its address should be supplied. It means that at run-time, a routine must be entered every time a parameter is referred to, to supply

its address. This leads to considerable inefficiency and it is rather difficult to construct.

As far as the FTN compiler is concerned the type of call is by no means a call by name. Appendix 2.1 proves this statement when optimization level one is requested. Examining the COMPASS expansion we see that:

- registers X5,X4,X3,X2 are loaded with the addresses of X,X,W and Z using the pointer to the stack, provided in AØ.
- the initial value of X (value 1), is loaded into X1.
- the new value for X (value 2) is computed in X7 and the memory location reserved for X is updated.
- when Z is computed, the old value of X, existing in X1, is used so that the result 4=1+3 is obtained.

In appendix 2.2 we see that when optimization level zero is requested, the same procedure is followed, but for X it is used the value in memory address and not the one in register; so that the result 5=2+3 is obtained.

A summary of values to be expected for the program in example # 1 for the four possible types of parameter passing mechanisms is presented in table 2.

TYPE OF CALL	VALUES/EXAMPLE 1		
	X	Y	Z
Call by reference	2.0	2.0	5.0
Call by value	1.0	2.0	7.0
Call by value/ result	2.0	2.0	4.0
Call by name	2.0	2.0	6.0

TABLE 2

CONCLUDING REMARKS

-1. The call by name is by far the best method of parameter passing since it gives no side effects; it is the most diffi-

cult to implement and eventually leads to inefficiency as far as execution time is concerned. It has not been encountered in any of the compilers under scrutinity.

-2. As far as the unprejudiced programmer is concerned, he must not take for granted whatever manuals state, but he must try to understand what lies behind each type of call and he must be able to test the compiler he is using.

References

1. FORTRAN Extended Reference Manual. CDC publication #60329100.
2. FORTRAN Reference Manual. CDC publication #60174900.
3. FORTRAN Reference Manual. IBM publication.
4. G.J.Myers. IBM Systems Journal, vol. 15, #3, 1976.

Received by Publishing Department
on March 30, 1977.

APPENDIX 1. Tests made with the FORTRAN F compiler.

FORTRAN main program and its ASSEMBLER expansion.

The subroutine SUB in FORTRAN and its ASSEMBLER expansion

1. When call by value result is used (APPENDIX 1.1)
2. When call by reference is used (APPENDIX 1.2)

```
DOS FORTRAN IV 360N-FO-479 3.8          MAINPGM          DATE  27

0001                X=1.0
0002                Y=2.0
0003                Z=7.0
0004                CALL SUB(X,X,X+Y,Z)
0005                WRITE(3,1) X,Y,Z
0006          1     FORMAT(1H ,3F9.1)
0007                STOP
0008                END

DOS FORTRAN IV 360N-FO-479 3.8          MAINPGM          DATE  27

LOCATION    STA NUM      LABEL    OP          OPERAND
000000                            BALR        15,0
000002                            LM          2,3,34(15)
000006                            L           13,30(0,15)
00000A                            LA          15,2(0,15)
00000E                            ST          15,4(0,13)
000012                            BCR         15,2
000014                    A4      DC          00000000
000018                            DC          97D4C1C9
00001C                            DC          D5D7C7D4
000020                            DC          00000000
000024                            DC          00000000
000028                            DC          00000000
00002C                    A52     L           12,4(0,13)
000030                            L           14,12(0,13)
000034                            LM          2,12,28(13)
000038                            MVI         12(13),255
00003C                            BCR         15,14
000040                    A36     L           15,112(0,13)
000044                            LR          12,13
000046                            LR          13,4
000048                            BAL         14,64(0,15)
00004C                            LR          13,12
000050          1                 LE          0,204(0,13)
000054                            STE         0,96(0,13)
000058          2                 LE          0,208(0,13)
00005C                            STE         0,100(0,13)
000060          3                 LE          0,212(0,13)
000064                            STE         0,104(0,13)
000068          4                 LE          0,96(0,13)
00006C                            AE          0,100(0,13)
000070                            STE         0,216(0,13)
000074                            LA          1,116(0,13)
000078                            L           15,108(0,13)
00007C                            BALR        14,15
00007E                            BC          0,4(0,0)
000082          5                 L           15,112(0,13)
000086                            BCR         0,0
000088                            BAL         14,4(0,15)
00008C                            DC          00000003
000090                            DC          000000B0
000094                            L           15,112(0,13)
000098                            BAL         14,8(0,15)
00009C                            DC          0470D060
0000A0                            BAL         14,8(0,15)
0000A4                            DC          0470D064
0000A8                            BAL         14,8(0,15)
0000AC                            DC          0470D068
0000B0                            BAL         14,16(0,15)
0000B4          7                 L           15,112(0,13)
0000B8                            BAL         14,52(0,15)
0000BC                            DC          05404040
0000C0                            DC          4040
                                  END
 TOTAL MEMORY REQUIREMENTS 0019A BYTES
```

APPENDIX 1.1. The subroutine SUB in FORTRAN and its ASSEMBLER expansion produced by the FORTRAN F compiler, when standard (call by value/result) parameter passing is requested.

```
0001              SUBROUTINE SUB(A,B,C,D)
0002              B=A+A
0003              D=A+C
0004              RETURN
0005              END
```

```
DOS FORTRAN IV 360N-FC-479 3-8          SUB                    DATE

                        SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATI
B               90    A               94    D               98

DOS FORTRAN IV 360N-FC-479 3-8          SUB                    DATE

LOCATION   STA NUM    LABEL    OP          OPERAND
000000                          BC          15,12(0,15)
000004                          DC          07E2E4C2
000008                          DC          40404040
00000C                          STM         14,12,12(13)
000010                          LM          2,3,40(15)
000014                          LR          4,13
000016                          L           13,36(0,15)
00001A                          ST          13,8(0,4)
00001E                          STM         3,4,0(13)
000022                          BCR         15,2
000024                          DC          00000000
000028                          DC          00000000
00002C                          DC          00000000
000038                  A20     L           2,0(0,1)
00003C                          MVC         100(3,13),0(2)
0000E2                          L           2,4(0,1)
0000E6                          MVC         96(3,13),0(2)
0000EC                          L           2,8(0,1)
0000F0                          MVC         108(3,13),0(2)
0000F6                          L           2,12(0,1)
0000FA                          MVC         104(3,13),0(2)
000100                          BALR        2,0
000102                          L           3,6(0,2)
000106                          BCR         15,3
000108                          DC          00000000
00010C                  A36     L           1,4(0,13)
000110                          L           1,24(0,1)
000114                          L           2,0(0,1)
000118                          MVC         0(3,2),100(13)
00011E                          L           2,4(0,1)
000122                          MVC         0(3,2),96(13)
000128                          L           2,8(0,1)
00012C                          MVC         0(3,2),108(13)
000132                          L           2,12(0,1)
000136                          MVC         0(3,2),104(13)
00013C                          L           13,4(0,13)
000140                          L           14,12(0,13)
000144                          LM          2,12,28(13)
000148                          MVI         12(13),255
00014C                          BCR         15,14
00014E          2       A52     LE          0,100(0,13)
000152                          AE          0,100(0,13)
000156                          STE         0,96(0,13)
00015A          3               LE          0,100(0,13)
00015E                          AE          0,108(0,13)
000162                          STE         0,104(0,13)
000166          4               SR          15,15
000168                          L           1,0(0,13)
00016C                          BCR         15,1
                                END
 TOTAL MEMORY REQUIREMENTS 00016E BYTES
```

The results obtained in this case:

X = 2.0 Y = 2.0 Z = 4.0

Indeed a call by value results is used.

APPENDIX 1.2. The subroutine SUB and its ASSEMBLER expansion when call by reference is requested by the user.

```
0001                SUBROUTINE SUB(/A/,/B/,/C/,/D/)
0002                B=A+A
0003                D=A+C
0004                RETURN
0005                END
```

```
DOS FORTRAN IV 360N-FO-479 3.8          SUB                    DATE

                        SCALAR MAP
SYMBOL      LOCATION    SYMBOL      LOCATION    SYMBOL      LOCATIC
B                 90    A                 94    D                 98

DOS FORTRAN IV 360N-FO-479 3.8          SUB                    DATE

LOCATION    STA NUM     LABEL     OP          OPERAND
000000                            BC          15,12(0,15)
000004                            DC          07E2E4C2
000008                            DC          40404040
00000C                            STM         14,12,12(13)
000010                            LM          2,3,40(15)
000014                            LR          4,13
000016                            L           13,36(0,15)
00001A                            ST          13,8(0,4)
00001E                            STM         3,4,0(13)
000022                            BCR         15,2
000024                            DC          00000000
000028                            DC          00000000
00002C                            DC          00000000
000030                A20         L           2,0(0,1)
000034                            LA          2,0(2,0)
000038                            ST          2,100(0,13)
00003C                            L           2,4(0,1)
000040                            LA          2,0(2,0)
000044                            ST          2,96(0,13)
000048                            L           2,8(0,1)
00004C                            LA          2,0(2,0)
000050                            ST          2,108(0,13)
000054                            L           2,12(0,1)
000058                            LA          2,0(2,0)
00005C                            ST          2,104(0,13)
000060                            BALR        2,0
000062                            L           3,6(0,2)
000066                            BCR         15,3
000068                            DC          00000000
000070                A36         L           1,4(0,13)
000074                            L           1,24(0,1)
000078                            L           13,4(0,13)
00007C                            L           14,12(0,13)
000080                            LM          2,12,28(13)
000084                            MVI         12(13),255
000088                            BCR         15,14
00008A              2   A52       L           10,100(0,13)
00008E                            LE          0,0(0,10)
000092                            AE          0,0(0,10)
000096                            L           11,96(0,13)
00009A                            STE         0,0(0,11)
00009E              3             LE          0,0(0,10)
0000A2                            L           12,108(0,13)
0000A6                            AE          0,0(0,12)
0000AA                            L           10,104(0,13)
0000AE                            STE         0,0(0,10)
0000B2              4             SR          15,15
0000B4                            L           1,0(0,13)
0000B8                            BCR         15,1
                                  END
 TOTAL MEMORY REQUIREMENTS 0000BA BYTES
```

The results obtained in this case:

X = 2.0 Y = 2.0 Z = 5.0

They indicate that it is really a call by reference.

APPENDIX 2. Tests with the FORTRAN Extended - FTN compiler.

Appendix 2.1. Main program and subroutine COMPASS expansion when OPT=1.

Appendix 2.2. The same but for the case OPT= 0

```
AM START      73/74   OPT=1                                   FTN 4.6+428


      PROGRAM START (INPUT,OUTPUT)
      X=1.0
      Y=2.0
      Z=7.0
      CALL SUB(X,X,X+Y,Z)
      PRINT 1,X,Y,Z
    1 FORMAT(1H ,3F15.1)
      STOP
      END
004144 DATA.                                          CON. BSS 0B
004144 DATA.      1720400000000000000                   DATA 1720400000000000003
004145 DATA.      1721400000000000000                   DATA 1721400000000000003
004146 DATA.      1722700000000000000                   DATA 1722700000000000003
004147 DATA.      0000000000000000000                   DATA 0000000000000000003
                                                        EXT    STOP.
                                                        EXT    OUTCI.
                                                        EXT    SUB
                                                        EXT    QBNTRY.
004150 DATA.                                          X       BSS 1B
004151 DATA.                                          Y       BSS 1B
004152 DATA.                                          Z       BSS 1B
                                                      *         USE   CODE.
                                                                                LI
004113 CODE.     5150004144           DATA.                     SA5   CON.
                        5140004145    DATA.                     SA4   CON.+1B
004114 CODE.     5130004146           DATA.                     SA3   CON.+2B
                        5110004125    CODE.                     SA1   FAP1
004115 CODE.     10755                                          BX7   X5
                       300+5                                    FX0   X4+X5
                         10644                                  BX6   X4
004116 CODE.     5170004150           DATA.                     SA7   X
                        24700                                   NX7   B0,X0
004117 CODE.     5160004151           DATA.                     SA6   Y
                         16633                                  BX6   X3
004120 CODE.     5170004132           CODE.                     SA7   ST.
                        5160004152    DATA.                     SA6   Z
004121 CODE.     0100000000           <EXT>    +                RJT   SUB,B3
                        0065004111
004122 CODE.     5110004133           DATA.                     SA1   FIO1
004123 CODE.     0100000000           <EXT>    +                RJT   OUTCI.,B0
                        0066004111
004124 CODE.     5110004147           DATA.                     SA1   CON.+3B
                        0400000000    <EXT>                     EQ    STOP.
004125 CODE.                                          FAP1      BSS   0B
004125 CODE.     0000000000000004150  DATA.                     APL   X
004126 CODE.     0000001000000004150  DATA.                     APL   X
004127 CODE.     0000000000000004132  CODE.                     APL   ST.
004130 CODE.     0000000000000004152  DATA.                     APL   Z
004131 CODE.     0000001000000000000                            APL
004132 CODE.                                          ST.       BSS   1B
                                                      Z.        END   START
```

APPENDIX 2.1. The subroutine SUB in FORTRAN and its COMPASS expansion when the user requested optimization level one /OPT=1/ to the FTN.

```
)TINE SUB        73/74   OPT=1                                  FTN 4.6+428

      SUBROUTINE SUB(A,B,C,D)
      B=A+A
      D=A+C
      RETURN
      END
```

COMPASS code as generated by the FTN compiler :

```
                                                    USE DATA.
                                                    USE START.
000000 START.   2325025555555500000003              TRACE   SUB,SUB,4B
000001 START.   0000000000000000000000
000002 START.   5130001001520300000000              ENTRY   SUB,ENTRY.,0,0
000003 START.   0410400003010004550O
000004 START.   7460054010516000000001
                                                    FORPAR  A
                                                    FORPAR  B
                                                    FORPAR  C
                                                    FORPAR  D
                                                    USE DATA..
                                                    USE DATA.
                                                            USE     CODE.
*
000005 CODE.    54500                                   SA5     A0
                     5040000002                         SA4     A0+2B
000006 CODE.    5030000001                              SA3     A0+1B
                          5020000003                    SA2     A0+3B
000007 CODE.    53150                                   SA1     X5
                     30711                              FX7     X1*X1
                          53540                         SA5     X4
                               53730                    SA7     X3
000010 CODE.    30051                                   FX0     X5*X1
                     24000                              NX6     B0,X0
                          53620                         SA6     X2
000011 CODE.    0400000002              START.          EQ      EXIT.
                                                  Z.    END
```

The results obtained in this case (they indicate a call by value/result parameter passing mechanism):

X = 2.0 Y = 2.0 Z = 4.0

The FORTRAN Extended Reference manual[1] states that the call by name constitutes the standard parameter passing mechanism, associated with a call by value for expressions appearing as actual parameters. Differences between different optimization levels in this respect are not mentioned.

APPENDIX 2.2. The subroutine SUB in FORTRAN and its COMPASS expansion when the user requested optimization level zero /OPT=0/.

```
SUB        73/74   OPT=0  TRACE                    FTN 4.6+428

      SUBROUTINE SUB(A,B,C,D)
      C=A+A
      D=A+C
      RETURN
      END
```

COMPASS code as generated by the FTN compiler

```
                                          USE DATA.
                                          USE START.
000000 START.  2325022555555550000004     TRACE   SUB,SUB,48
000001 START.  0000000000000000000
000002 START.  5130000001520300000        PENTRY  SUB,ENTRY.,3,1
000003 START.  6102777763610200108
000004 START.  0400401034610046000
000005 START.  7400540105160000001
                                          FORPAR  A
                                          FORPAR  B
                                          FORPAR  C
                                          FORPAR  D
                                          USE DATA..
                                          USE DATA.
                                                  USE     CODE.
                                          *
000006 CODE.   6102000002                         SB0     B2+2B
                     54500                        SA5     A0
                         53450                    SA4     X5
000007 CODE.   30744                              FX7     X4*X4
                   5036000001                     SA3     A0+1B
                         53730                    SA7     X3
                                          *
000010 CODE.   6102000003                         SB0     B2+3B
                     54500                        SA5     A5
                         53450                    SA4     X5
000011 CODE.   5030000002                         SA3     A0+2B
                     5020000003                   SA2     A0+3B
000012 CODE.   53530                              SA5     X3
                   30054                          FX0     X5+X4
                     24700                        NX7     B0,X0
                         53720                    SA7     X2
                                          *
000013 CODE.   6102000004                         SB0     B2+4B
                       0400000002 START.          EQ      EXIT.
                                          Z.      END
```

The results obtained in this case (they indicate a call by reference, parameter passing mechanism):

X = 2.0 Y = 2.0 Z = 5.0

```
      PROGRAM START (INPUT,OUTPUT)
      X=1.0
      Y=2.0
      Z=7.0
      CALL SUB(X,X,X+Y,Z)
      PRINT 1,X,Y,Z
    1 FORMAT(1H ,3F15.1)
      STOP
      END
```

```
004151 DATA.
004151 DATA.      1720400000000000000
004152 DATA.      1721400000000000003
004153 DATA.      1722700000000000007
004154 DATA.      0000000000000000000
004155 DATA.
004156 DATA.
004157 DATA.

004114 CODE.      6102000002
                        5150004151 DATA.
004115 CODE.      10755
                       5170004155      DATA.
004116 CODE.      6102000003
                        5150004152 DATA.
004117 CODE.      10755
                       5170004156      DATA.
004120 CODE.      6102001004
                        5150004153 DATA.
004121 CODE.      10755
                       5170004157      DATA.
004122 CODE.      6102001005
                        5150004155 DATA.
004123 CODE.      5140004156      DATA.
                        5110004132 CODE.
004124 CODE.      30045
                     20700
                       5170004137 CODE.
004125 CODE.      0100001000           <EXT>
                        0005004111
004126 CODE.      6102000006
                        5110004140 DATA.
004127 CODE.      0100000000           <EXT>
                        0006004111
004130 CODE.      6102000010
                        5110004154 DATA.
004131 CODE.      0400001000           <EXT>
004132 CODE.
004132 CODE.      0000001000004155 DATA.
004133 CODE.      0000000000004155 DATA.
004134 CODE.      0010000000004137 CODE.
004135 CODE.      0000000000004157 DATA.
004136 CODE.      0060000000000000000
004137 CODE.
```

```
CON.  BSS  0B
      DATA 1720400000000000003
      DATA 1721400000000000003
      DATA 1722700000000000007
      DATA 0000000000000000000
      EXT    STOP.
      EXT    OUTCI.
      EXT    SUB
      EXT    FTNRPV.
X     BSS  1B
Y     BSS  1B
Z     BSS  1B
      USE    CODE.
*
      SB0    B2+2B
      SA5    CON.
      BX7    X5
      SA7    X
*
+     SB0    B2+3B
      SA5    CON.+1B
      BX7    X5
      SA7    Y
*
+     SB0    B2+4B
      SA5    CON.+2B
      BX7    X5
      SA7    Z
*
+     SB0    B2+5B
      SA5    X
      SA4    Y
      SA1    IAP1
      FX0    X4+X5
      NX7    B0,X0
      SA7    ST.
+     RJT    SUB,5B
*
      SB0    B2+6B
      SA1    JIO1
+     RJT    OUTCI.,6B
*
      SB0    B2+1C9
      SA1    CON.+3B
      EQ     STOP.
IAP1  BSS    0B
      APL    X
      APL    X
      APL    ST.
      APL    Z
      APL
ST.   BSS    1B
Z.    END    START
```