



Title	FÖRTRAN 700の特徴について
Author(s)	後藤, 米子
Citation	大阪大学大型計算機センターニュース. 1973, 10, p. 94-102
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65196">https://hdl.handle.net/11094/65196</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# FÖRTRAN 700 の特徴について

研究開発室 後藤 米子

当センターでFÖRTRAN700 を使用しはじめてからおよそ1年になりますが、利用者の方にFÖRTRAN700 の特徴を知っておいていただいた方が、何かと便利なのではないかと思われまますので、以下「プログラム単位の結合について」「DIMENSTOL やCÖMMÖNで宣言した領域を荒した場合について」実際にどうなるかということを具体例をあげて説明してみたいと思います。以下、簡単のために、FÖRTRAN700, FORTRANL-500 のことをF-700, FL-500 と書きます。

## 1. プログラム単位の結合について

FÖRTRAN700 コンパイラーが、コンパイル（翻訳）時に、外部手続きに関して出力している情報は

- ① 手続きの型
- ② 引数の数

の2点です。

①は、プログラミング説明書のP、108の<sup>(表1)</sup>ように、分類されます。

②は、引数の並びの数です。引用するプログラム単位内と、手続きプログラムの中で①、②が一致していないときには、リンク時にエラーメッセージ LLK66I TYPE/ADCON UNMATCHサブプログラム名が出力されます。

又、外部手続き名を引用しているのに、システム側でサーチした結果、見つからない場合には、エラーメッセージ

```
LLK 38I UNRESÖLVED SUBPRÖGRAM
      17I UNRESÖLVED ENTRIES
```

が出力されます。これらの他に、プログラム単位の結合時に出力されるエラー・メッセージとして、資料(i)のようなものがあります。

注意していただきたい点は、F-700では、対応する引数の型が一致するかどうかチェックされていないという点です（資料(ii)のTEST4）すなわち、実引数の内部表現が、そのまま、仮引数の内部表現として読まれますので、出力結果がどうもおかしいと、首をかしげることになるでしょう。（FL-500では、型の変換は自動的に行われます。）

但し、F-700で、実引数が倍精度実数型、仮引数が単精度実数型の場合、正常に実行されまます（資料(iii)）。実引数と仮引数の対応は、JIS7000では、表2のようになっています。

\*) は、JIS7000では定義されていませんが、F-700では、配列の先頭の要素だけが対応します(文法P.122)(資料(ii)TEST3)。

\*\*\*) は、JIS 7000, F-700ともに許されていませんが(文法P.121), エラーメッセージは、出力されずに、そのまま実行に入ります(資料(iv)PRCD1)。

仮引数を再定義している場合、対応する実引数は、変数名、配列要素名、配列名でなければなりません。ループの中で実引数として定数を書くと問題になります(資料(iv)PRCD2)。

多重entry文を使用するときには、SUBROUTINE宣言した手続き名を、必ず引用しておかなければなりません(資料(ii)ISN5)。さもない場合には、ENTRY文で宣言した手続き名が結合されません。(LLK 38I, 17Iが出力される)。

2. COMMON, DIMENSIONで宣言、確保した領域を荒すことについて COMMON, DIMENSIONで宣言、確保した領域(データ領域)をこえて書き込みにいった場合、自分のプログラムを自分で荒すことになります。FORTRANコンパイラーは、FORTRAN700で記述された原始プログラムをプログラム単位毎に翻訳していきませんが、番地の低い方から表3のように割り当てられています。データ領域をこえて書きこみしていく場合は、実行時のあらゆるエラー状態が起ります。このプログラムは、わざわざ、エラー状態を起したものですが、ICHECKを250番地から5語しか(資料V)確保していないのに isn 5で、ICOUNT=6となったとき、ICHECK(6)として、320番地のMAXの値を、6として書きかえてしまいますので、isn 10のIF文を満足してしまい、文番号1に実行の制御がわたります。

次にisn 14で、IC=16になりますが、isn 16でICHECK(16)として440番地のSPR出力の6を17と書きかえることになりしますので、isn 17のWRITE文を実行しようとしたときに、140Fのエラーが出力されます。命令や番地をデータで書き換えてしまいますとそれぞれ

```
MRMOOI UEP OP-CODE ERP=04
```

```
MRMOOI UEP ADDRESS ERR=10
```

のエラー・メッセージが出力されます。上記のメッセージが出たときには、SUBCHKで、だいたいわかります。デバック文は、資料(iv)のような要領で使います。

### 3. 実行時におけるO除算及びオーバーフローについて

#### 3-1. 整数型データ以外のO除算、及びオーバーフローについて

既にセンターニュースNo.9 P. 33, P. 44にも書きましたが、OPTIONOVFを使うことによって、実行時のCPU-TIMEは増加しますが、400N, 401N のメッセージが出力されます。又、基本外部 サブルーチンOVERFL, DVCHKによっても検出できます。

O除算の結果は、資料(vi)で示しますように、場合によって、Oになったり(C(1)~C(4))被

除数の値 (C(5)) になつたりしますのでご注意下さい。

### 3-2. 整数型データの場合

OPTIONOVE, OVERFL, DVCHK のいずれによつても、0除算及びオーバー・フローを検出することができませんのでご注意下さい。

表 1

P <sub>1</sub>	意 味
0	サブルーチン手続き
1	文字型関数手続き
2	論理型関数手続き
3	整数型関数手続き
4	実数型関数手続き
5	倍精度実数型関数手続き
6	複素数型関数手続き
7	倍精度複素数型関数手続き

表 2

仮引数	変数名	配列名
実引数		
変 数 名	○	×***)
配列要素名	○	○
配 列 名	—*)	○
その他の式	○	×

#### (i) LLK67I \*\*\*\*Labeled Common Size Error

(カードイメージ)

説 明：先に定義された名前付共通ブロックのサイズより後から定義された同じ名前の名前付共通ブロックのサイズが大きい。(プログラム・レコード：04)

#### LLK68I \*\*\*\*Duplicate Subprogram Name

(カード イメージ)

説 明：主プログラムの名前、副プログラムの名前 (サブルーチン、副プログラム、関数副プログラム、初期値設定副プログラム) と名前付共通ブロックの名前と同じものがある。(プログラム・コード：20)

現在の運用ではリンク時コンディションカードで、実行をバイパスするようにしています。

```
(ii) DIMENSION A(5),C(3)
      DATA A/1.0,2.0,3.0,4.0,5.0/,C/0.1,0.2,0.3/
      DATA X,Y/1.0,0.20.0/,I/2/
      CALL TEST
      CALL TEST1 (A, C, X, Y)
      WRITE (6,130) A, C, X, Y
130  FORMAT(1H0,10F5,1)
      CALL TEST2 (A(1),C(2),X, Y)
      WRITE (6,130) A, C, X, Y
      CALL TEST3 (A, C, X, Y)
      WRITE (6,130) A, C, X, Y
      WRITE (6,133) I
```

```

133  FORMAT(1H0, 10 HBEFORE I=, I3)
      CALL TEST4(I)
      WRITE(6, 132)  I
132  FORMAT(1H0, 3H I=, B48)
      WRITE(6, 134)  I
134  FORMAT(1H0, 10 HAFTER I=, I3)
      STOP
      END
      SUBROUTINE TEST
      DIMENSION AA(5),CC(3),E(1)
      RETURN
      ENTRY TEST1 (AA, E, XX, YY)
      WRITE(6, 140) AA, E, XX, YY
140  FORMAT(1H0, 5HTEST1/1H , 5F5.1/1H , 3F5.1/1H , 2F5.1
      RETURN
      ENTRY TEST2 (AA, CC, XX, YY)
      WRITE(6, 141) AA, CC, XX, YY
141  FORMAT(1H0, 5HTEST2/1H , 5F5.1/1H , 3F5.1/1H , 2F5.1)
      DO 10 K=1, 3
10   CC(K)=FLOAT(K)
      RETURN
      ENTRY TEST3 (AA, P, X, Y)
      WRITE(6, 142) AA, P, X, Y
142  FORMAT(1H0, 5HTEST3/1H , 5F5.1/1H , 3F5.1)
      P=3.3
      RETURN
      ENTRY TEST4 (Q)
      WRITE(6, 143) Q
143  FORMAT(1H0, 5HTEST4/1H , 3H Q=,B48)
      WRITE(6, 144) Q
144  FORMAT(1H0, 3H Q=,F5.1)
      RETURN
      END

TEST1
      1.0 2.0 3.0 4.0 5.0 .1 10.0 20.0
      1.0 2.0 3.0 4.0 5.0 .1 .2 .3 10.0 20.0

TEST2
      1.0 2.0 3.0 4.0 5.0 .2 .3 10.0 10.0 20.0
      1.0 2.0 3.0 4.0 5.0 .1 1.0 2.0 3.0 20.0

TEST3
      1.0 2.0 3.0 4.0 5.0 .1 3.0 20.0
      1.0 2.0 3.0 4.0 5.0 3.3 1.0 2.0 3.0 20.0

```



```

SUBROUTINE PRCDR
  DIMENSIONX1(1)
  ENTRY PRCDR1 (X1, X2, X3, X4)
  WRITE(6, LIST) X1, X2, X3, X4
  DO 5 I=1, 3
5 X1(I)=FLOAT(I)
  RETURN
  ENTRY PRCDR2 (Y1, Y2, Y3)
  Y3 =Y3+1.0
  RETURN
END

```

```

**SUBTRACE 5 BEFORE PRCDR
**SUBTRACE 5 AFTER PRCDR
**SUBTRACE 6 BEFORE PRCDR1
**SUBTRACE          ARGO1 =11.000000000      :260000000000000004
**SUBTRACE          ARGO2 =12.000000000      :300000000000000004
**SUBTRACE          ARGO3 =13.000000000      :320000000000000004
**SUBTRACE          ARGO4 =14.000000000      :340000000000000004
  11.00000000    12.00000000    13.00000000    14.00000000
**SUBTRACE 6 AFTER PRCDR1
**SUBTRACE          ARGO1 =1.000000000        :200000000000000001
**SUBTRACE          ARGO2 =2.000000000        :200000000000000002
**SUBTRACE          ARGO3 =2.999999999        :300000000000000002
**SUBTRACE          ARGO4 =14.000000000       :340000000000000004
  1.00          2.00          3.00          14.00
**SUBTRACE 9 BEFORE PRCDR2
**SUBTRACE          ARGO1 =1.000000000        :200000000000000001
**SUBTRACE          ARGO2 =2.000000000        :200000000000000002
**SUBTRACE          ARGO3 =2.999999999        :300000000000000002
**SUBTRACE 9 AFTER PRCDR2
**SUBTRACE          ARGO1 =1.000000000        :200000000000000001
**SUBTRACE          ARGO2 =2.000000000        :200000000000000002
**SUBTRACE          ARGO3 =4.000000000        :200000000000000003
**SUBTRACE 9 BEFORE PRCDR2
**SUBTRACE          ARGO1 =1.000000000        :200000000000000001
**SUBTRACE          ARGO2 =2.000000000        :200000000000000002
**SUBTRACE          ARGO3 =4.000000000        :200000000000000003
**SUBTRACE 9 AFTER PRCDR2
**SUBTRACE          ARGO1 =1.000000000        :200000000000000001
**SUBTRACE          ARGO2 =2.000000000        :200000000000000002
**SUBTRACE          ARGO3 =5.000000000        :240000000000000003

```

```

**SUBTRACE 9 BEFORE PRCDR2
**SUBTRACE          ARGO1 =1.0000000000      :2000000000000001
**SUBTRACE          ARGO2 =2.0000000000      :2000000000000002
**SUBTRACE          ARGO3 =5.0000000000      :2400000000000003
**SUBTRACE 9 AFTER PRCDR2
**SUBTRACE          ARGO1 =1.0000000000      :2000000000000001
**SUBTRACE          ARGO2 =2.0000000000      :2000000000000002
**SUBTRACE          ARGO3 =5.9999999998      :3000000000000003
**SUBTRACE 9 BEFORE PRCDR2
**SUBTRACE          ARGO1 =1.0000000000      :2000000000000001
**SUBTRACE          ARGO2 =2.0000000000      :2000000000000002
**SUBTRACE          ARGO3 =5.9999999998      :3000000000000003
**SUBTRACE 9 AFTER PRCDR2
**SUBTRACE          ARGO1 =1.0000000000      :2000000000000001
**SUBTRACE          ARGO2 =2.0000000000      :2000000000000002
**SUBTRACE          ARGO3 =6.9999999999      :3400000000000003

```

ISN

```

(v) 0001          DATA MAX, ICOUNT, IC/100, 0, 5/
0002          DIMENSION ICHK(5)
0003          2 WRITE(6,10) ICHK
0004          10 FORMAT(1HO, 5I5)
0005          ICOUNT=ICOUNT+1
0006          I=1COUNT
0007          ICHK(I)=I
0008          WRITE(6,11) ICHK(I)
0009          11 FORMAT(1HO, I5)
0010          IF(ICOUNT-MAX) 2, 1, 1
0011          1 WRITE(6,12) ICOUNT, MAX
0012          12 FORMAT(1HO, 2I5)
0013          MMAX=50
0014          3 IC=IC+11
0015          WRITE(6,11) IC
0016          ICHK(IC)=17
0017          WRITE(6,11) ICHK(IC)
0018          IF(IC-MMAX) 3, 4, 4
0019          4 STOP
0020          END

```

オブジェクト・メモリ・マップ (OPTIONMAPで出力される)

NAME	OWORDS	NAME	TP	ADDRESS	WORDS	NAME	TP	ADDRESS	WORDS	NAME	TP	ADDRESS	WORDS
NON COMMON ARRAYS:		ICLK	I	0000250	000005	1COUNT	I	0000330	000001	1C	I	0000340	000001
VARIABLES: MAX			I	0000320	000001	MMAX	I	0000360	000001				
			I	0000350	000001								

オブジェクト・リスト

45		CLST				0000150				0000507	00		
		COMNT											DIMENSION VECTOR
45	00	0000230	ADCON	00	00	00	0000250						
15	00	0000234	CONST				00000005						
15	00	0000240	CONST				07777770						
15	00	0000244	CONST				00000010						
		COMNT											DATA STATEMENT
15	00	0000320	CONST				00000000000000144	.....					MAX
15	00	0000330	CONST				00000000000000000	.....					I COUNT
15	00	0000340	CONST				00000000000000005	.....					I C
		COMNT											FORMAT STATEMENT
15	00	0000370	CONST				1000001400010015000531000501						
15	00	0000406	CONST				1000001400010031000501						
15	00	0000421	CONST				1000001400010015000231000501						
		COMNT											CONSTANT DATA
15	00	0000440	CONST				00000000000000006						
15	00	0000450	CONST				00000000000000001						
15	00	0000460	CONST				00000000000000062						
15	00	0000470	CONST				00000000000000013						
15	00	0000590	CONST				00000000000000021						
04	00	0000117	ADCON	00	00	00	0000510						
		COMNT					00002						WRITE STATEMENT

出力結果 0 0 0 0 0

1

1 0 0 0 0

\*\*140F WRITE-ILLEGAL FILE REFERENCE  
NUMBER 17

2

1 2 0 0 0

ERROR DETECTED IN NFSBISQWTF  
00036244

3

1 2 3 0 0

CALLED FROM MAINPG

4

0017 00004504

1 2 3 4 0

\*\*\*\*\*MRMOO1 UEP 0007712 DEPENDENT

5

1 2 3 4 5

6

6 6

16

表 3

プロローグ
(エピローグ)
dummy argument
dimension vector
array
variable
namelist
format
constant
statement function
procedure

プログラム単位を結合した  
場合には

level Common
メイン
サブ
:
:
NFSANCHOR
object
関数
blank Common

```
(vi)  DIMENSION C(5), J(5)
      DATA A, B, X, Y / 0, 0, 1.0, 2.0, 0.0 /
      C(1)=B/A
      C(2)=5.0 / (1.0-1.0)
      C(3)=B / (3.0-3.0) +FUNC(X)
      C(4)=FUNC(X) / A + FUNC(Y)
      C(5)=FUNC(X) / (X+Y*X*Y-X)
      WRITE(6,LIST) C
      STOP
      END
      FUNCTION FUNC(X)
      FUNC=X**2
      END
      .0      .0      4.000000000      .0      4.000000000
```