

## FORDAPシステム利用に関する補足

牛島, 和夫  
九州大学工学部情報工学科

藤村, 直美  
九州大学大学院工学研究科

<https://doi.org/10.15017/1472521>

---

出版情報：九州大学大型計算機センター広報. 10 (1), pp.5-9, 1977-03-01. 九州大学大型計算機センター  
バージョン：  
権利関係：

## FORDAP システム利用に関する補足

牛島 和夫\* 藤村 直美\*\*

## 1. ソースエレメントが複数個の場合の処理

大記憶にファイルされているソースプログラムを FORDAP にかけてその実行状態を解析するには

```

¥NO
¥USER
¥QJOB
¥FORDAPC SOURCE=FILE
¥POFILE READ,F0039.SLIB(SC1)
¥FORDAPL
¥FORDAPGO



デ ー タ



¥JEND

```

とすれば良い(文献(1)P. 118 参照)。ただし、ソースプログラムはファイル名 F0039.SLIB 中の SC1 というエレメントとして登録してある。

現在の FORDAP システムではこのエレメントが 1 個しか指定できないようになっている。ところが、特に CPS 等で大記憶ファイルを使ってプログラムを作成・実行しようとする時には、ソースプログラムを幾つかのエレメントに分割してファイルに保存しておいた方が都合の良いことがある。このような場合、そのままでは FORDAP システムによる計測に不都合が生ずることがある。ここでは大記憶ファイル中に分割して保存されているソースエレメントを FORDAP システムの計測の対象とするための方法について簡単に述べる。要は複数個のソースエレメントをまとめて一つにしようというものである。ファイル局所名 FORDAP 中の二つのソースエレメント S. MAIN と S. SUBR をまとめて一つのファイルあるいはエレメントとする場合を例にして説明する。

## (1) CPS による方法

CPS の OLD, SAVE コマンドを用いれば複数個のエレメントを一つにまとめることができる。もちろんこれは保存ファイルの領域にそれだけの余裕がある時に限られる。S. MAIN と S. SUBR の IDENT が異なる時には図 1 ( a ) のようにすれば良い。これでファイル FORDAP 中に S. MAIN の後に S. SUBR を結合した新しいエレメント S. BOTH が作成できたことになる。

---

\* 九州大学工学部情報工学科  
\*\* 九州大学大学院工学研究科博士課程

SWPRV	FORDAP	SWPRV	FORDAP
OLD	S. MAIN	OLD	S. MAIN
OLD, DA	S. SUBR	OLD, DA	S. SUBR, 1000, 10
SAVE	S. BOTH	SAVE	S. BOTH
( a )		( b )	

図1. CPSによる例

一方、S. MAINとS. SUBRのIDENTが同じ場合にはS. SUBRの行番号をS. MAINの行番号の最大値より大きい値で始まるように変更する必要がある。例えばS. MAINの行番号の最大値が990であれば図1(b)のようにする。なお詳しくは文献(2) pp. 68~69を参照されたい。こうして作成されたソースエレメントは例えば図2に示す制御文によってFORDAPシステムによる計測が実現できる。(CPSのBATCHコマンドを利用するため¥USER文は省略、以下同じ。)

```

*NC      A0039010,KEEP
*QJOB
*FORDAPC SOURCE=FILE
*POFILE  READ,F0039.FORDAP(S,BOTH)
*FORDAPL
*FORDAPCO
*JEND
    
```

図2.

(2) プログラムによる方法

ファイルに余裕がなくて先述の方法が使えない時など一時的ファイルを利用することができる。そのためには複数個のエレメントをまとめて一つの一時的ファイルを作成するプログラムがあれば良い。図3にそのプログラムリストを示す。このプログラムは結合したいエレメントを各々マルチファイルの要素として次々に入力し、一時的ファイルに出力していくものである。最後にマルチファイルとしてのファイル定義名がない状態になると、すなわち結合したいエレメントがなくなると、標準では「FT612 MISSING FD CARD」というエラーメッセージが出力され異常終了するので、FORTRAN Hのエラーモニタ機能を用いてこの出力を抑制し、みかけ上正常終了させている。

S. MAINとS. SUBRを結合してFORDAPシステムにかけるための制御文は図4のようになる。ここでE. CONCATは、図3のプログラムに与えた実行形式プログラム名(EBNAME)である。なお複数個のエレメントをまとめたファイルは保存されないでFORDAPシステムを利用するたびにこのプログラムを実行する必要がある。そのためE. CONCATは予め作成して登録しておけばよい。その作成手順の一例を図5に示した。ここでS. CONCATは図3のソースエレメント名である。

```

C      MAIN PROGRAM FOR ELEMENT CONCATENATION
C
C      EXTERNAL EOF
C
C      CALL ERRSET(612,2,-1,1,EOF)
C
C      CALL CRCOPY
C
C      STOP
C
C      END

C      SUBROUTINE CRCOPY
C
C      INTEGER CR(20)
C
C      NIN=1
C      NOUT=2
C
C      100 READ(NIN,10,END=100,ERR=200) CR
C          WRITE(NOUT,10) CR
C          GO TO 100
C
C      200 WRITE(6,20)
C          RETURN
C
C      ENTRY EOF
C
C      STOP
C
C      10 FORMAT(20A4)
C      20 FORMAT(10X,12HABNORMAL END )
C
C      END

```

## 図 3

```

*NO      A0039010,*KEEP
*JOB
*RUN     EBNAME=E.CONCAT,EBFLNAME=F0039.FORDAP,LPFIL=1
*POFILE F01.001,F0039.FORDAP(S.MAIN)
*POFILE F01.002,F0039.FORDAP(S.SUBR)
*WKFILE F02,FILENAME=SOURCE,FCBPRM=YES,RCDSIZE=80,BLKSIZE=1200,
        TMOD=9,DISP=PASS
*FORDAPC SOURCE=FILE
*WKFILE  READ,FILENAME=SOURCE
*FORDAPL
*FORDAPGO
*JEND

```

## 図 4

```

#SWPRV FORDAP
#FORTRANH
#ATTACH SOURCE,FORDAP
#RUN,CL 'NOLIST,AREA=10,N0ARGCHK,0PT2,E=(S.CONCAT) '
. プログラム メイ: FORTRANH, カンリョウ コード: 000.
. プログラム メイ: LIED , カンリョウ コード: 000.
#PUT EXELIB,EXQTPRGM
#RENAME EXQTPRGM,E.CONCAT
#DETACH SOURCE
#RESET

```

## 図 5

## 2. 主プログラムを解析対象から除く方法

ソースプログラムの一部だけ FORDAP にかける場合次のようにする (文献(1) p. 118 参照)。

¥NO

¥USER

¥QJOB

¥FORTRANH

FORDAP システムにかけない ソースプログラムカード
---------------------------------

... 主プログラムおよび STOP 文を含むプログラム  
単位を含んではならない。

¥FORDAPC

FORDAP で解析したい ソースプログラムカード
------------------------------

... 主プログラムを必ず含むこと。  
STOP を含むプログラム単位は必ずここに含ま  
れていること。

¥FORDAPL

¥FORDAPGO

デ - タ
-------

¥JEND

ここで FORDAP にかけたいソースプログラム中には必ず主プログラムと STOP を含むプログラム単位を含まなければならない。STOP 文に行きついた所で FORDAP システムが事後処理を行って解析結果のリストを出力するためである。

ところが主プログラムの解析は不要だという場合がよく生じる。このような時は主プログラムそのものを SUBROUTINE にかえてやればよい。すなわち、主プログラム中の STOP 文を全て RETURN 文に変えて頭に例えば

SUBROUTINE DUMMY

とつけて STOP 文を含まない副プログラムに変更する。そして

CALL DUMMY

STOP

END

という主プログラムを新しく作って実行解析の対象となる別の副プログラムとこれとを FORDAP にかければ良い。なお FORDAP で解析を必要としない部分は予め相対形式にしておくともっと手間が省ける。上例の SUBROUTINE DUMMY を相対形式エレメント DUMMY としておいて FORDAP を使用するには次のようにする (文献(1) p. 117 参照)。

```

¥NO
¥USER
¥QJOB
¥FORDAPC
  CALL DUMMY
  STOP
  END

```

FORDAPで解析したい ソースプログラムカード
-----------------------------

```

¥FORDAPL  SYSINPRM=YES
NAME EXQTPRGM,ENTRY=ELM(MAIN)
CALL  FORTLIBH,F. SSLH , P. LIB
SELECT RELBIN
SELECT PRVLIB(DUMMY,SUB1,SUB2)
FIN
¥POFILE PRVLIB,F0039. PLIB
¥FORDAPGO

```

デ ー タ
-------

```
¥JEND
```

ここでSUB1,SUB2もFORDAPで解析を必要としないものですでに相対形式となっているエレメント名である。

#### 参 考 文 献

1. 牛島；FORDAP—FORTRANプログラム実行解析システム—について，九大大型計算機センター広報Vol. 9, No. 2(1976)
2. 九州大学大型計算機センター；利用の手引—会話型処理編