FORTRAN77プログラム動的解析ツール : FORDAP77の使用について

牛島, 和夫
九州大学工学部情報工学科

程, 京徳
九州大学工学部情報工学科

https://doi.org/10.15017/1472564

出版情報：九州大学大型計算機センター広報. 17 (4/5), pp.276-287, 1984-09-25. 九州大学大型計算機センター
バージョン：
権利関係：

研 究 開 発

FORTRAN77プログラム動的解析ツール －FORDAP77－の使用について

牛島　和夫*，程　京徳*

1. はじめに

FORTRANプログラム動的解析ツールFORDAPは1975年九州大学工学部情報工学科のFACOM 230−45Sのもとで作成以来，九大大型計算機センターはじめ50以上の大学研究機関等に移し換えられて利用されてきた[1][2]．1982年2月にFORTRAN77がJIS化された[3]にもかかわらずFORDAPは新しい規格に対応していなかった．1984年5月から九大大型計算機センターのFORTRAN処理系がFORTRAN77に統一されることになったのでFORTRAN77プログラムを解析対象とする動的解析ツール（以下これをFORDAP77という）を用意することにした．FORDAP77は九州大学大型計算機センターのFACOM M−382 OSⅣ/F4のもとで使用できる．これは従来のFORDAPに基づいてFORDAPの全面改訂を行ったものである．本稿では，2.でFORDAP77の機能について述べ，3.でFORDAP77の使用方法について述べ，4.でFORDAP77の使用上の制限および注意事項について述べる．

2. FORDAP77の機能

FORDAP77の主な機能は次のとおりである．

(1) FORTRAN77プログラムの各実行文の実行回数を計測する．論理IF文，ブロックIF文及びELSE IF文ではさらに論理式の値が真となる場合の数を計測する．

(2) プログラムの実行コストとして各プログラム単位の実行所要CPU時間（計測によるオーバーヘッドが含まれる）を計測する．

(3) 実行時に得られた計測データをプログラム単位ごとに整理，集計してサマリーを作る．

(4) 上記の結果をわかりやすい形で（TSSの場合では利用者の指示によって会話的に）出力する．

(5) プログラミングデバッグ支援のためにプログラム異常終了時にもそれまで得られた情報を可能な限り出力する．

FORDAP77の出力について次に説明する．

FORDAP77は計測結果全体を一覧できるように，最初にサマリー（上記(3)）を出力する．サマリーの例を図1に示す．FORDAP77をTSSで使用する場合にはサマリーを見て問題のある手続き（があればそれ）を見付け出し，その計測結果をすぐに見にゆくときなどの手掛りとなる．

サマリーの中の各欄の意味は次のとおりである（図1参照）．

・ NAME欄：プログラム単位の名前．名前の前にMとかFとかSとかDとかがあり，それぞれは該当プログラム単位が主プログラム，関数副プログラム，サブルーチン副プログラム，初期値設定副プログラムであることを示す．

---

* 九州大学工学部情報工学科

```
FACOM   OSIV/F4      FORDAP77      -840225-      V-03   L-01        DATE 84-05-08     TIME 17-53-23

                ********* SUMMARY OF PROCEDURES IN SOURCE PROGRAM *********

   NAME    *  NO * LINES * EXECUTABLE * UNEXECUTED * INVOKED * EXECUTIONS *  TIME

M FDP77        1     129          18            0         0         4793     7106
F CKIND        2      18           8            0     15850        62331      692
S DOST         3      71          44           15         4          190        6
S DOTERM       4      30          12            0        98          202        2
S IFCONV       5      44          27            8        14          240       48
S IFST         6      84          55           14       183        32942      644
S INIT         7      16           7            0        31          186        0
S INPUT        8      27           7            0      1789         8943      996
F IUP          9      31           8            3       842         3368       44
S LGEXP       10      25          13            2        81          329       86
F NUM         11      16           0            1       419         3659       20
S NXTCH       12      26          14            0     14758       290543      693
S NXTST       13      83          49            7      1195        58543     1219
S OUTBIF      14      32          14            0       112         1100      203
S OUTBN       16      13           5            0       542         1930       50
S OUTBI       17       7           2            0      1476         1476      233
S OUTCRN      18      28          11            0       844         5154      687
S OUTCR1      19      10           2            0      1129         1129      482
S OUTMSF      15      50          33           14        31          192        3
S OUT0        20      52          20            6       153          408       86
S OUT1D       21      39          11            2        27           80       11
S OUT1S       22      35          13            2       786         2411      335
S OUT2D       23      11           2            0        98           98       51
S OUT3R       24      32          15            5       181          643       76
S PARSER      25     390         242           94      1195       103871     3653
S STACK       26      20           8            3      8271        33084      347
S STFNC       27      24          11            9         7            7        0
S TIMSET      28      17           7            0        31          186       14
S TITLE       29      16           2            0         1            1       10
S TRANS       30     323         202           40      1195        21414     1932
D ******      31     101           1            0         0            0        0

*****TOTAL*****     1800         871          225     51343       639533
```

図　1.

・　NO欄：ソース・プログラムの中におけるプログラム単位の出現順序.

・　LINES欄：プログラム単位の総行数.

・　EXECUTABLE欄：プログラム単位の中の実行文の総数. 各IF文とGO　TO文の枝の分岐も実行文の総数に入れる.

・　UNEXECUTED欄：プログラム単位の実行文の中で実行されなかった文の数.

・　INVOKED欄：プログラム単位が引用された回数.

・　EXECUTIONS欄：プログラム単位の各実行文の実行回数の総数.

・　TIME欄：プログラム単位の実行所要CPU時間(MSEC).

各プログラム単位のサマリーの中での出現順序はまず主プログラム(名前を付けない場合は＊＊＊＊＊＊)，次に関数副プログラムとサブルーチン副プログラム(名前のアルファベット順)，最後に初期値設定副プログラム(名前のアルファベット順．名前を付けない場合は＊＊＊＊＊＊)となっている.

FORDAP77リストの形式には二種類がある. 図2と図3とにFORDAP77の二通りの出力例を示す. NXTSTというサブルーチンは図1のサマリー中に見出すことができる.

```
* SOURCE STATEMENT *                EXECUTION TIME =        1219 MSEC          EXECUTIONS          TRUE
                                                                                    (1)                (2)
      C                                                                      00005180
            SUBROUTINE NXTST                                                 00005190          1195
      C                                                                      00005200
      C     THIS SUBROUTINE READS CARDS UP TO NEXT STATEMENT.                00005210
      C                                                                      00005220
            INTEGER CREND,CRNO,CMTNO                                         00005230
            INTEGER TIMW,TIMV,IDCL,MODE                                      00005240
            CHARACTER WORK*30,CHAR*1,CRBUF(21)*80                            00005250
            INTEGER IEND,ZERO,NBIF,NLIF,NEIF,NMA,NSF,NBD                     00005260
            INTEGER LINK(21),FIRST,LAST                                      00005270
            INTEGER ITIME                                                    00005280
      C                                                                      00005290
            COMMON /FCOM2/CREND,CRNO,CMTNO                                   00005300
            COMMON /FCOM4/TIMW,TIMV,IDCL,MODE                                00005310
            COMMON /FCOM5/WORK,CHAR,CRBUF                                    00005320
            COMMON /CNSTBN/IEND,ZERO,NBIF,NLIF,NEIF,NMA,NSF,NBD              00005330
            COMMON /CRLINK/LINK,FIRST,LAST                                   00005340
            COMMON /CPUTIM/ITIME                                             00005350
            COMMON /IOUNIT/NIN,NOUT,NTBL,NMSG,NCR                            00005360
      C                                                                      00005370
            IF (CREND.GT.0) THEN                                             00005380          1195          0 (  0.0% )
      C                                                                      00005390
      C     LAST END CARD IS MISSED.                                         00005400
      C                                                                      00005410
               WRITE(NMSG,20)                                                00005420             0
               CALL OUTB(7)                                                  00005430             0
               STOP                                                          00005440             0
            ENDIF                                                            00005450          1195
            CMTNO=0                                                          00005460          1195
            IF (CRNO.NE.0) THEN                                              00005470          1195       1194 ( 99.9% )
               FIRST=LAST                                                    00005480          1194
               CRNO=1                                                        00005490          1194
               ELSE                                                          00005500             1
      C                                                                      00005510
      C     NO CARD EXISTS IN BUFFER                                         00005520
      C                                                                      00005530
               CRNO=1                                                        00005540             1
               FIRST=1                                                       00005550             1
               LAST=1                                                        00005560             1
               DO 50 I=1,20                                                  00005570             1
         50    LINK(I)=I+1                                                   00005580            20
               LINK(21)=1                                                    00005590             1
               CALL INPUT(CRBUF(FIRST))                                      00005600             1
        100    IF (CRBUF(FIRST)(1:1).EQ.'C'.OR.CRBUF(FIRST)(1:1).EQ.'*') THEN 00005610           2          1 ( 50.0% )
                  IF (TIMW.EQ.0) CALL TIMSET                                 00005620             1          1 (100.0% )
                  CALL OUTB1(ZERO)                                           00005630             1
                  IF (CRBUF(FIRST)(2:6).EQ.'-TIME') ITIME=0                  00005640             1          0 (  0.0% )
                  CALL INPUT(CRBUF(FIRST))                                   00005650             1
                  GO TO 100                                                  00005660             1
               ENDIF                                                         00005670             1
            ENDIF                                                            00005680          1195
        200 CRNO=CRNO+1                                                      00005690          1229
            IF (CRNO.GT.21) THEN                                             00005700          1229          0 (  0.0% )
      C                                                                      00005710
      C     BUFFER OVER                                                      00005720
      C                                                                      00005730
               WRITE(NMSG,10) CRBUF                                          00005740             0
               STOP                                                          00005750             0
            ENDIF                                                            00005760          1229
            LAST=LINK(LAST)                                                  00005770          1229
        300 CALL INPUT(CRBUF(LAST))                                          00005780          1787
            IF (CREND.GT.0) RETURN                                           00005790          1787          1 (  0.1% )
            IF (CRBUF(LAST)(1:1).EQ.'C'.OR.CRBUF(LAST)(1:1).EQ.'*') THEN     00005800          1786        558 ( 31.2% )
               CMTNO=CMTNO+1                                                 00005810           558
               GO TO 300                                                     00005820           558
               ELSE                                                          00005830          1228
               I=1                                                           00005840          1228
        400    IF (I.LT.72.AND.CRBUF(LAST)(I:I).EQ.' ') THEN                 00005850         10871       9643 ( 88.7% )
                  I=I+1                                                      00005860          9643
                  GO TO 400                                                  00005870          9643
               ENDIF                                                         00005880          1228
               IF (I.EQ.72) THEN                                             00005890          1228          0 (  0.0% )
                  CMTNO=CMTNO+1                                              00005900             0
                  GO TO 300                                                  00005910             0
               ENDIF                                                         00005920          1228
            ENDIF                                                            00005930          1228
            IF (CRBUF(LAST)(6:6).NE.'0'.AND.                                 00005940          1228         34 (  2.8% )
           +    CRBUF(LAST)(6:6).NE.' ') GO TO 200                           00005950
      C                                                                      00005960
         10 FORMAT(//5X,27HTOO MANY CONTINUATION LINES/(5X,A80))             00005970
         20 FORMAT(//5X,18HEND CARD IS MISSED )                              00005980
      C                                                                      00005990
            END                                                              00006000
```

図 2.

```
EXECUTIONS ** SOURCE STATEMENT **          EXECUTION TIME =        1196 MSEC
           C
  (1)  1195        SUBROUTINE NXTST
           C
           C       THIS SUBROUTINE READS CARDS UP TO NEXT STATEMENT.
           C
                   INTEGER CREND,CRNO,CMTNO
                   INTEGER TIMW,TIMV,IDCL,MODE
                   CHARACTER WORK*30,CHAR*1,CRBUF(21)*80
                   INTEGER IEND,ZERO,NBIF,NLIF,NEIF,NMA,NSF,NBD
                   INTEGER LINK(21),FIRST,LAST
                   INTEGER ITIME
           C
                   COMMON /FCOM2/CREND,CRNO,CMTNO
                   COMMON /FCOM4/TIMW,TIMV,IDCL,MODE
                   COMMON /FCOM5/WORK,CHAR,CRBUF
                   COMMON /CNSTBN/IEND,ZERO,NBIF,NLIF,NEIF,NMA,NSF,NBD
                   COMMON /CRLINK/LINK,FIRST,LAST
                   COMMON /CPUTIM/ITIME
                   COMMON /IOUNIT/NIN,NOUT,NTBL,NMSG,NCR
           C
      1195         IF (CREND.GT.0) THEN
         0(  0.0%)
           C
           C       LAST END CARD IS MISSED.
           C
         0            WRITE(NMSG,20)
         0            CALL OUT0(7)
         0            STOP
      1195         ENDIF
      1195         CMTNO=0
      1195         IF (CRNO.NE.0) THEN
  (2) 1194( 99.9%)
      1194             FIRST=LAST
      1194             CRNO=1
         1             ELSE
           C
           C       NO CARD EXISTS IN BUFFER
           C
         1             CRN0=1
         1             FIRST=1
         1             LAST=1
         1             DO 50 I=1,20
        20     50      LINK(I)=I+1
         1             LINK(21)=1
         1             CALL INPUT(CRBUF(FIRST))
         2    100      IF (CRBUF(FIRST)(1:1).EQ.'C'.OR.CRBUF(FIRST)(1:1).EQ.'*') T
         1( 50.0%)                                                                 HEN
         1                IF (TIMW.EQ.0) CALL TIMSET
         1(100.0%)
         1                CALL OUTB1(ZERO)
         1                IF (CRBUF(FIRST)(2:6).EQ.'-TIME') ITIME=0
         0(  0.0%)
         1                CALL INPUT(CRBUF(FIRST))
         1                GO TO 100
         1             ENDIF
      1195         ENDIF
      1229    200  CRNO=CRNO+1
      1229         IF (CRNO.GT.21) THEN
         0(  0.0%)
           C
           C       BUFFER OVER
           C
         0            WRITE(NMSG,10) CRBUF
         0            STOP
      1229         ENDIF
      1229         LAST=LINK(LAST)
      1787    300  CALL INPUT(CRBUF(LAST))
      1787         IF (CREND.GT.0) RETURN
         1(  0.1%)
      1786         IF (CRBUF(LAST)(1:1).EQ.'C'.OR.CRBUF(LAST)(1:1).EQ.'*') THEN
       558( 31.2%)
       558             CMTNO=CMTNO+1
```

図　3.

```
  558           GO TO 300
 1228           ELSE
 1228           I=1
10871   400    IF (I.LT.72.AND.CRBUF(LAST)(I:I).EQ.' ') THEN
 9643( 88.7%)
 9643              I=I+1
 9643              GO TO 400
 1228           ENDIF
 1228           IF (I.EQ.72) THEN
    0(  0.0%)
    0              CMTNO=CMTNO+1
    0              GO TO 300
 1228           ENDIF
 1228        ENDIF
 1228        IF (CRBUF(LAST)(6:6).NE.'0'.AND.
   34(  2.8%)
           +     CRBUF(LAST)(6:6).NE.' ') GO TO 200
      C
         10 FORMAT(//5X,27HTOO MANY CONTINUATION LINES/(5X,A80))
         20 FORMAT(//5X,18HEND CARD IS MISSED )
      C
            END
```

図　3.（つづき）

図２はバッチ処理向きの出力形式で，出力幅の広い装置（１３２欄）を想定して計測結果を出力するものである．図３はTSS向きの出力形式で，出力幅の狭い装置（８０欄）を想定して計測結果を出力するものである．図２においては，ソース・テキストの右側にあるEXECUTIONS欄（図２①参照）には対応する実行文の実行回数を示し，EXECUTIONS欄の右側にあるTRUE欄（図２②参照）には論理ＩＦ文，ブロックＩＦ文，ELSE　ＩＦ文の論理式を評価して真となった回数を百分率とともに示している．図３においては，出力幅が狭いので図２①，②の情報を図の左側に示し，②の情報は次の行に出力している（図３②参照）．

3. FORDAP 77の使用方法

FORDAP77を使用するには，利用者はいくつかの簡単なジョブ制御文或はTSSコマンドを使用しさえすればよい．

(1)　バッチ処理で使用する場合

バッチ処理で使用する例は図４に示す．自明だと思われるので説明しない．

```
// jobname    JOB password
//            EXEC FORDAP77
// FDP.SYSIN DD DSN= dataset name, DISP=SHR
// GO.SYSIN  DD *
   data
//
```

図　4.

(2)　TSSで使用する場合

FORDAP77コマンドは次のとおりである．

FORDAP77　ソース・プログラム・データセット名
OBJ（結合したいオブジェクト・モジュール・データセット名）
LIB（結合したいロード・モジュール・データセット名）
DATA（入力データが入っているデータセット名）
RESULT（実行結果を出力するデータセット名）
SY（出力装置の指定）
TYPE(FORDAP77リスト出力形式の指定）
TERMINAL（端末出力行数の指定）

各パラメータをそれぞれ次に説明する．

OBJ(　)の指定は，ソース・プログラムとコンパイル済みのオブジェクト・モジュールとを結合して実行するためのものである．もしそのようなオブジェクト・モジュールがなければOBJ(　)指定を省略できる．

LIB(　)指定は，コンパイル・リンケージ済みのロード・モジュールを結合して実行するためのものである．もしそのようなロード・モジュールがなければLIB(　)指定を省略できる．

DATA(　)指定は，かっこの中で「＊」或は「入力データを格納しているデータセット名」を指定する．前者は入力データを端末から入力することを示す．後者は入力データを該当データセットから入力するのを示す（プログラムの中でのREAD文の論理機番は「5」であること）．もし省略すると前者と見なす．

RESULT(　)指定は，かっこの中で「＊」或は「実行結果を格納したいデータセット名」を指定する．前者は実行結果を端末に出力することを示す．後者は実行結果を該当データセットに格納することを示す（プログラムの中でのWRITE文の論理機番は「6」であること）．もし省略すると前者と見なす．

SY(　)指定は，かっこの中で「＊」或は「A」，「D」，「O」，「S」，「K」，「U」のいずれかを指定する．前者はTSS端末出力で，後者はセンターのラインプリンタへの出力である．もし省略するとTSS端末に出力する．

TYPE(　)指定は，かっこの中で「N」或は「W」を指定する．前者は80欄用で，後者は132欄用を想定している．指定を省略すると80欄となる．

TERMINAL(　)指定により，端末への出力の一時停止を指定できる．かっこの中に「＊」或は「整数」を書く．前者は通常の方法で端末を使うことを示す．後者は「整数」で指定した行数だけ端末にメッセージが出力されたとき，出力を停止して擬似アテンション割込み要求の有無を，端末ユーザに問い合わせるよう指定する．実際には，利用者はもう一度RETURN鍵を押すとメッセージが続いて出力される．この指定を省略すると前者になる．

TSSの場合の具体的な使用例を図5に示す．以下図中に丸で囲んだ番号に従って説明する．

FORDAP77コマンドを入力すると（図5①参照），FORDAP77システムはまず被計測ソース・プログラムの前処理を行う（図5②参照）．もしソース・プログラムの中でFORDAP77使用制限（次節に述べる）に違反することがあればFORDA77の使用制限に違反するメッセージもここで出力

する．その後，前処理をしたソース・プログラムはコンパイル・リンケージ・ロードされる（図5③，④参照），被計測プログラムが実行に入ると

```
***** EXECUTING *****
```

というメッセージが出てくる（図5⑤参照）．この時，もしデータを端末から入力する要求があれば利用者はデータを入力する（図5⑥参照）．その後，実行結果が出力される（図5⑦参照）．

プログラムの実行が終るとFORDAP77システムはまずFORDAP77の標題とソース・プログラムのサマリーを出力する（図5⑧参照）．次に

```
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
```

というメッセージが出る．ここから，利用者とFORDAP77との会話をはじめることができる．この状態を以下FORDAP77の会話モードという（図5⑨参照）．

FORDAP77は利用者のために三つのコマンドを用意している．以下それぞれ説明する．

・LIST：このコマンドは，利用者がソース・プログラムの中で任意の1つ或はいくつかのプログラム単位の計測情報を見ることができるように用意している．出力すべきプログラム単位の選定にはサマリーが参考になるだろう．このコマンドの入力形式は次のとおりである．

$$\left\{\begin{array}{l}\text{LIST}\sqcup\\ \text{L}\sqcup\end{array}\right\}\left\{\begin{array}{l}\text{文字列}\\ \text{整数}\\ \text{整数1}-\text{整数2}\\ *\end{array}\right.\left[\left[\left\{\begin{array}{l},\\ \sqcup\end{array}\right\}\left\{\begin{array}{l}\text{文字列}\\ \text{整数}\\ \text{整数1}-\text{整数2}\end{array}\right\}\right]\cdots\right\}$$

［注］

ここで「$\sqcup$」は1個以上の空白を示す．上記の中で，文字列は出力すべきプログラム単位の名前である（図5⑩参照）．整数は，ソース・プログラムの中でのプログラム単位の出現順序である（図5⑪参照）．それぞれサマリーの中でNAME欄とNO欄に示されている．整数1－整数2の場合は利用者がソース・プログラムの中で連続する2個以上のプログラム単位について計測情報を見るために用意している．この場合には整数1と整数2は必ずソース・プログラムの中でのプログラム単位の出現順序で，かつ整数1≦整数2でなければならない（図5⑫参照）．なお，一つのLISTコマンドの中でいくつかの文字列と整数と整数1－整数2とが混在してもかまわない（図5⑬参照）．＊は，ソース・プログラム全体の計測情報を出力するためのものである（図5⑭参照）．

・SUMMARY：このコマンドは，利用者とFORDAPとの会話中いつでもプログラム単位ごとのサマリーを見ることができるように用意している．このコマンドの入力形式は次のとおりである．

$$\left\{\begin{array}{l}\text{SUMMARY}\sqcup\\ \text{S}\sqcup\end{array}\right\}\left[\begin{array}{l}\text{A}\\ \text{N}\end{array}\right]$$

［注］

ここでオプション「A」と「N」はサマリー出力形式の指定である．前者はプログラム単位の名前のアルファベット順でサマリーを出力するのを示す．後者はプログラム単位の出現順でサマリーを出力するのを示す．この指定を省略すると「A」指定と見なす．FORDAP77の会話モードのもとで利用

者がSUMMARYコマンドを入力すると何度でもサマリーを出力することができる（図5⑮，⑯参照）．

```
READY
FORDAP77 TESTMFSD.FORT77 (1)
    ***** FORDAP77 PREPROCESSOR ENTERED ***** (2)
    ***** END OF FORDAP77 PREPROCESSOR *****
FORTRAN 77 COMPILER ENTERED
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(Z      ),FLAG(E),OPTIMIZE(2)
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(L      ),FLAG(E),OPTIMIZE(2)
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(MFSD   ),FLAG(E),OPTIMIZE(2) (3)
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(A      ),FLAG(E),OPTIMIZE(2)
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(F1     ),FLAG(E),OPTIMIZE(2)
FORTRAN 77 ERROR MESSAGES: PROGRAM NAME(F2     ),FLAG(E),OPTIMIZE(2)
END OF COMPILATION
** MEMBER NAME ** WORK     NOW ADDED TO LIBRARY. (4)
    ***** EXECUTING ***** (5)
-- PROGRAM MFSD ENTERED
-- INPUT DATA(I5) PLEASE
(6)12345
**    12345+      113=    12458 (7)
-- END OF PROGRAM MFSD


FACOM  OSIV/F4  FORDAP77  -840225-  V-03  L-01  DATE 84-05-14  TIME 18-13-27


          ********* SUMMARY OF PROCEDURES IN SOURCE PROGRAM *********  (8)

  NAME    *  NO * LINES * EXECUTABLE * UNEXECUTED * INVOKED * EXECUTIONS *  TIME

M MFSD        6       17          10            0         0            9       9
S A           7        5           2            0         1            1       0
F F1          9        4           2            0         1            1       0
F F2         10        4           2            0         1            1       0
S L           4        5           2            0         1            1       1
S Z           2        5           2            0         1            1       0
D ******      3        5           1            0         0            0       0
D ******      5        5           1            0         0            0       0
D DX          8        5           1            0         0            0       0
D XYZ         1        6           1            0         0            0       0

*****TOTAL*****      61          24            0         5           14

***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END ) (9)
(10)L MFSD
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          9 MSEC
           *
                 PROGRAM MFSD
                 COMMON /D/IZYX,ID1,ID2,IDX
           1        WRITE(6,10)
              10 FORMAT('--- PROGRAM MFSD ENTERED'/'--- INPUT DATA(I5) PLEASE')
           1        READ(5,20) I
              20 FORMAT(I5)
           1        CALL Z
           1        CALL L
           1        CALL A
           1        IF=ID2+F1(10)+F2(100)
           1        WRITE(6,25) I,IF,I+IF
              25 FORMAT('***',I8,'+',I8,'=',I8)
           1        WRITE(6,30)
              30 FORMAT('--- END OF PROGRAM MFSD')
           1        STOP
                    END
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
(11)L 1
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *     THIS IS PROGRAM FOR TEST MFSD

                 BLOCK DATA XYZ
                 COMMON /D/IZYX,ID1,ID2,IDX
                 DATA IZYX/10/
                 END
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
(12)L 3-4
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
```

図　5.

```
                    BLOCK DATA
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA ID1/100/
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          1 MSEC
           *
         1          SUBROUTINE L
                    COMMON /D/IZYX,ID1,ID2,IDX
         1          IL=ID1+1
                    END
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
(13)L    8    F2   2-3
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
                    BLOCK DATA DX
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA IDX/300/
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
         1          FUNCTION F2(IP)
         1          F2=IP+2
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
         1          SUBROUTINE Z
                    COMMON /D/IZYX,ID1,ID2,IDX
         1          IZ=IZYX+1
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
                    BLOCK DATA
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA ID1/100/
                    END
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
(14)L *
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *     THIS IS PROGRAM FOR TEST MFSD

                    BLOCK DATA XYZ
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA IZYX/10/
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
         1          SUBROUTINE Z
                    COMMON /D/IZYX,ID1,ID2,IDX
         1          IZ=IZYX+1
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
                    BLOCK DATA
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA ID1/100/
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          1 MSEC
           *
         1          SUBROUTINE L
                    COMMON /D/IZYX,ID1,ID2,IDX
         1          IL=ID1+1
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          0 MSEC
           *
                    BLOCK DATA
                    COMMON /D/IZYX,ID1,ID2,IDX
                    DATA ID2/200/
                    END
EXECUTIONS ** SOURCE STATEMENT **        EXECUTION TIME =          9 MSEC
           *
                    PROGRAM MFSD
                    COMMON /D/IZYX,ID1,ID2,IDX
         1          WRITE(6,10)
        10 FORMAT('--- PROGRAM MFSD ENTERED'/'--- INPUT DATA(I5) PLEASE')
         1          READ(5,20) I
        20 FORMAT(I5)
         1          CALL Z
         1          CALL L
         1          CALL A
         1          IF=ID2+F1(10)+F2(100)
         1          WRITE(6,25) I,IF,I+IF
```

図　5.（つづき）

```
            25 FORMAT('***',I8,'+',I8,'=',I8)
      1        WRITE(6,30)
            30 FORMAT('--- END OF PROGRAM MFSD')
      1        STOP
               END
EXECUTIONS ** SOURCE STATEMENT **         EXECUTION TIME =          0 MSEC
           *
      1        SUBROUTINE A
               COMMON /D/IZYX,ID1,ID2,IDX
      1        IA=IDX+1
               END
EXECUTIONS ** SOURCE STATEMENT **         EXECUTION TIME =          0 MSEC
           *
               BLOCK DATA DX
               COMMON /D/IZYX,ID1,ID2,IDX
               DATA IDX/300/
               END
EXECUTIONS ** SOURCE STATEMENT **         EXECUTION TIME =          0 MSEC
           *
      1        FUNCTION F1(IP)
      1        F1=IP+1
               END
EXECUTIONS ** SOURCE STATEMENT **         EXECUTION TIME =          0 MSEC
           *
      1        FUNCTION F2(IP)
      1        F2=IP+2
               END
***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
S A
```

(15)

```
         ********* SUMMARY OF PROCEDURES IN SOURCE PROGRAM *********

  NAME    *  NO * LINES * EXECUTABLE * UNEXECUTED * INVOKED * EXECUTIONS *  TIME

M MFSD        6       17           10            0         0            9        9
S A           7        5            2            0         1            1        0
F F1          9        4            2            0         1            1        0
F F2         10        4            2            0         1            1        0
S L           4        5            2            0         1            1        1
S Z           2        5            2            0         1            1        0
D ******      3        5            1            0         0            0        0
D ******      5        5            1            0         0            0        0
D DX          8        5            1            0         0            0        0
D XYZ         1        6            1            0         0            0        0

*****TOTAL*****      61           24            0         5           14

***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
SUMMARY N
```

(16)

```
         ********* SUMMARY OF PROCEDURES IN SOURCE PROGRAM *********

  NAME    *  NO * LINES * EXECUTABLE * UNEXECUTED * INVOKED * EXECUTIONS *  TIME

D XYZ         1        6            1            0         0            0        0
S Z           2        5            2            0         1            1        0
D ******      3        5            1            0         0            0        0
S L           4        5            2            0         1            1        1
D ******      5        5            1            0         0            0        0
M MFSD        6       17           10            0         0            9        9
S A           7        5            2            0         1            1        0
D DX          8        5            1            0         0            0        0
F F1          9        4            2            0         1            1        0
F F2         10        4            2            0         1            1        0

*****TOTAL*****      61           24            0         5           14

***** INPUT COMMAND PLEASE ( SUMMARY , LIST , END )
```

(17)

```
END
     ***** END OF FORDAP77 SERVICE *****
READY
```

図　5.（つづき）

・END：このコマンドは利用者とFORDAP77との会話を終らせるためのものである．FORDAP77会話モードのもとで利用者が「END」或は「E」を入力すると会話が終る．このとき，FORDAP77は次の

```
***** END OF FORDAP77 SERVICE *****
```

というメッセージを出力する（図5⑰参照）．

4. FORDAP77の使用上の制限および注意事項

FORDAP77を使用する上で次の制限と注意事項がある．

(1) 計測の対象となるソース・プログラムは固定形式のものに限る．自由形式のものは，固定形式に変換した上でFORDAP77を使用すること．またこのソース・プログラムには主プログラムを必ず含まねばならぬ．

(2) 次の英字名は予約名となり利用者のプログラム中では使用できない．

SYS900, SYS901, SYS902, SYS903, SYS904,
SYS905, SYS906, SYS907, SYS908, SYS909

(3) 65000～65535の文番号は計測で使用しているので，利用者のソース・プログラムの中でこれらの文番号を使用できない．

(4) ファイル参照番号95, 96, 97, 98, 99を計測で使用しているので利用者はこれらを使用してはならない．

(5) FORDAP77は実行回数を計測するために計測用カウンタが必要である．カウンタの最高数は10000個になっている．このカウンタ数では経験的に約25000行程度のFORTRAN77プログラムが計測可能である．これで計測できない大きなFORTRAN77プログラムのときはFORDAP77の使用制限を違反するメッセージが出力される．

(6) FORDAP77は被計測ソース・プログラムのサブルーチン副プログラムと関数副プログラムとの合計数の上限を180個としている．初期値設定副プログラムの数の上限を20個としている．

(7) DO文とブロックIF文の入れ子の数の上限はFORTRAN JIS C6201(1982)[3]の中で定義していない．FORDAP77ではそれぞれの上限を25としている．

(8) FORDAP77の解析対象はFORTRAN JIS C6201(1982)[3]で書いたFORTRAN77プログラムであるが，今回，FORDAP77の実現に当ってJIS規格以外のFACOM OSⅣ/F4 FORTRAN77の仕様[4][5]も考慮に入れた．利用者は拡張されたDO文や各種のデバッグ文など[4]も使用することができる．しかし，FACOM OSⅣ/F4 FORTRAN77以外の拡張に対してはFORDAP77は正しく対処しないことがある．

(9) 元のFORDAPは入力されるFORTRANプログラムはあらかじめFORTRANコンパイラで文法違反がないことを確かめたものに限るという制限がある[2]．FORDAP77はこういう制限をとりはずした．したがってプログラム中に文法的な誤りがあればコンパイルする時エラー・メッセージが現われる．しかし，コンパイルされるプログラムはFORDAP77の前処理によってカウンタを挿入し

たものであるのでエラー・メッセージの行番号が元のソース・プログラムのものと違ってくるので注意されたい．

(10) 各プログラム単位ごとに計測される実行時間は各プログラム単位の入口と出口でそれまで使用されたCPU時間を計測してその差を積算したものである．実際のCPU時間の計測はFACOM OS Ⅳ/F4 FORTRAN77のCLOCKルーチンで行う．したがって計測の対象となる副プログラムが多数回呼ばれると（したがってCLOCKルーチンも多数回呼ばれると）そのためのオーバヘッドが無視できなくなる場合がある．このとき，次の対処の仕方がある．計測の対象となるFORTRAN77プログラムの一番先頭に注釈行として

C-TIME

を付け加えるとFORDAP77は主プログラムのみCPU時間を計測し副プログラムのCPU時間を計測しない．

5. おわりに

元のFORDAPは，FORTRANプログラムの改善を助けるツールとして，九大センターでよく使用していただいた．それがFORDAP77を作る原動力になった．FORDAP77を用いてプログラムを評価・改善する過程については，文献[1]を参照されたい．

FORDAP77の作成にあたって，元のFORDAPを実現した藤村直美助教授（九大情報処理教育センター）にはお世話になった．また，JOB制御文，コマンドプロシジャの作成，ユーザインタフェースの設計について，高木利久助手（九大情報工学科）から，さまざまな注意と指導を受けた．ここに記して謝意を表す．

参考文献

1. 牛島　FORTRANプログラミングツール，産業図書，1979年.
2. 藤村　FORTRANプログラム動的解析システム（FORDAPシステム）移し換え手引書，1981年.
3. JISハンドブック　情報処理，日本規格協会，1982年.
4. 計算機マニュアルFACOM OS Ⅳ FORTRAN77文法書（64SP－3330－3），富士通（株）.
5. 計算機マニュアルFACOM OS Ⅳ/F4 MSP FORTRAN77使用手引書V10用(78SP-5300-2)，富士通(株).

[注] { }は，その中に記述された複数個の項目の中から，一つだけ選択して指定することを示す．
[ ]は，その中に記述された項目を省略することができることを示す．
・・・は，その直前の項目を一回以上任意回繰り返して指定することができることを示す．