

COBOLプログラム輪郭作成システムの使用について

藤村, 直美
九州大学工学部情報工学科

牛島, 和夫
九州大学工学部情報工学科

<https://doi.org/10.15017/1472534>

出版情報 : 九州大学大型計算機センター広報. 11 (2), pp.72-79, 1978-06-01. 九州大学大型計算機センター

バージョン :

権利関係 :

COBOL プログラム輪郭作成システムの使用について

藤村 直美*・牛島 和夫*

1. はじめに

1976年3月から九大センターのFACOM 230-75 M-VII (1977年11月からはFACOM M-190 OS IV/F4)において使用に供されているFORTRANプログラム輪郭作成システム(FORDAP)^{1),2)}はFORTRANプログラムの各実行文の実行回数等を計数してプログラムの動的輪郭をフィードバックするものである。実行回数を計数するという簡単な原理に基づくシステムがプログラムの開発・改善の道具として有効なのは、第一に解くべき問題にかかわらず実行回数を通してプログラムを客観的に眺められること、第二にソースプログラムと得られた計測情報の対応付けが容易であること等のためと思われる。

FORTRANばかりでなく、他の手続き向き言語においてもこのようなシステムがプログラムの開発・改善の有効な道具になることが期待できる。そこでCOBOLに対しても同様な輪郭作成システム(以後COBOLDAPシステムと呼ぶ)を九州大学工学部情報工学科のFACOM 230-45 S OS IIの上に作成した。³⁾その後、このシステムは、他のいくつかの計算機システムに移し換えられ⁴⁾、東大大型計算機センターのHITAC 8800/8700 OS7⁵⁾、九大大型計算機センターのFACOM M-190 OS IV/F4でも利用可能となった。ここでは九大センターでCOBOLDAPシステムを使用する上での注意事項、制限、ジョブ制御文等について述べる。

2. COBOLDAP システム

COBOLDAPシステムは主につきのような機能を持っている。

- 1) COBOLプログラムの各実行文の実行回数を計数する。
- 2) 条件命令においては条件の成立回数を計数する。
- 3) 上記の結果をわかり易く出力する(図1参照)。

図1においてEXECUTIONS欄が各文の実行回数を示し、図の中程右側にある1(AT END)は左側に対応するREAD命令のAT END条件が一回成立したことを示す。SUBR SECTIONの実行回数が0回なのはこのセッションがPERFORM SUBRの形では実行されなかったことを示す。なお、1行に複数個の文を含む時も、各文の実行回数を正しく表示する。

このシステムの使用上の効果としてはつきのようなものがある。

- 1) 作成したプログラムの動作が客観的に認識できる。
- 2) 採用したアルゴリズムの適否が実行回数から判断できる。

* 九州大学工学部情報工学科

LINE	SEQ.	A.	B.	IDENT.	EXECUTIONS
FACOM M-190 OSIV/F4 CDBLDAP V-02 L-07 -780121- DATE 78.04.06 TIME 11:40:49 PAGE 0001					
0001	000010	IDENTIFICATION	DIVISION.		1
0002	000020	PROGRAM-ID.	CHRCNT.		
0003	000030	AUTHOR.	NAOMI FUJIMURA.		
0004	000040*				
0005	000050	ENVIRONMENT	DIVISION.		
0006	000060	CONFIGURATION	SECTION.		
0007	000070	SOURCE-COMPUTER.	FACOM-M190.		
0008	000080	OBJECT-COMPUTER.	FACOM-M190.		
0009	000090	SPECIAL-NAMES.	CO1 IS PAGE-CHANGE.		
0010	000100*				
0011	000110	INPUT-OUTPUT	SECTION.		
0012	000120	FILE-CONTROL.			
0013	000130	SELECT SOURCE-FILE ASSIGN TO S-SYSIN.			
0014	000140	SELECT PRINT-FILE ASSIGN TO S-SYSPRINT.			
0015	000150*				
0016	000160	DATA	DIVISION.		
0017	000170*				
0018	000180	FILE	SECTION.		
0019	000190	FD	SOURCE-FILE		
0020	000200		RECORDING MODE IS F		
0021	000210		BLOCK CONTAINS 0 RECORDS		
0022	000220		LABEL RECORD IS STANDARD.		
0023	000230	C1	CR-BUF.		
0024	000240		10 FILLER PIC X(180).		
0025	000250	FD	PRINT-FILE		
0026	000260		RECORDING MODE IS F		
0027	000270		BLOCK CONTAINS 20 RECORDS		
0028	000280		LABEL RECORD IS STANDARD.		
0029	000290	O1	LP-BUF.		
0030	000300		10 FILLER PIC X(123).		
0031	000310*				
0032	000320	WORKING-STORAGE	SECTION.		
0033	000330	77	EOF-SW PIC 9 VALUE IS 0.		
0034	000340	77	TBLCNT PIC 999 VALUE IS 1.		
0035	000350	77	TBLMAX PIC 999 VALUE IS 128.		
0036	000360	77	COLMAX PIC 999 VALUE IS 72.		
0037	000370	77	FOUND PIC 9.		
0038	000380	77	I PIC 999.		
0039	000390	77	J PIC 999.		
0040	000400	C1	CR.		
0041	000410		10 CHAR PIC X OCCURS 80 TIMES.		
0042	000420	O1	LP.		
0043	000430		10 FILLER PIC X(111) VALUE IS ' CHAR(*.		
0044	000440		10 LP-CHAR PIC X.		
0045	000450		10 FILLER PIC X(9) VALUE IS ') OCCURS *.		
0046	000460		10 LP-COUNT PIC Z(5)9.		
0047	000470		10 FILLER PIC X(7) VALUE IS ') TIMES*.		
0048	000480	O1	CHAR-TABLE.		
0049	000490		10 CH-TRL OCCURS 128 TIMES.		
0050	000500		20 CHAR-KIND PIC X.		
0051	000510		20 CHAR-COUNT PIC 9(6).		
0052	000520*				
0053	000530	PROCEDURE	DIVISION.		1
0054	000540*				
0055	000550	MAIN	SECTION.		1

LINE	SEQ.	A.	B.	IDENT.	EXECUTIONS
FACOM M-190 OSIV/F4 CDBLDAP V-03 L-07 -780121- DATE 78.04.06 TIME 11:40:49 PAGE 0002					
0056	000560*				
0057	000570	BEGIN.			1
0058	000580	PERFORM SHOKI-SETTEI.			1
0059	000590	PERFORM CARD-READ UNTIL EOF-SW = 1.			1
0060	000600	PERFORM ATO-SHIMATSU.			1
0061	000610*				
0062	000620	SUBR	SECTION.		0
0063	000630*				
0064	000640	SHOKI-SETTEI.			1
0065	000650	OPEN INPUT SOURCE-FILE OUTPUT PRINT-FILE.			1
0066	000660	MOVE SPACE TO LP-BUF.			1
0067	000670	WRITE LP-BUF AFTER PAGE-CHANGE.			1
0068	000680	PERFORM TABLE-CLEAR VARYING I FROM 1 BY 1			1
0069	000690	UNTIL I > TBLMAX.			1
0070	000700*				
0071	000710	TABLE-CLEAR.			128
0072	000720	MOVE SPACE TO CHAR-KIND (I).			128
0073	000730	MOVE ZERO TO CHAR-COUNT (I).			128
0074	000740*				
0075	000750	CARD-READ.			105
0076	000760	READ SOURCE-FILE INTO CR AT END MOVE 1 TO EOF-SW.			105
0077	000770	IF EOF-SW NOT = 1 PERFORM CHAR-CHECK VARYING I			105
0078	000780	FROM 1 BY 1 UNTIL I > COLMAX.			104
0079	000790*				
0080	000800	CHAR-CHECK.			7488
0081	000810	MOVE ZERO TO FOUND.			7488
0082	000820	PERFORM CHAR-COMPARE VARYING J FROM 1 BY 1			7488
0083	000830	UNTIL J > TBLCNT OR FOUND = 1.			
0084	000840	IF FOUND = 0			7488
0085	000850	ADD 1 TO TBLCNT			43
0086	000860	IF TBLCNT > TBLMAX PERFORM ATO-SHIMATSU			43
0087	000870	ELSE-ADD 1 TO CHAR-COUNT (TBLCNT)			43
0088	000880	MOVE CHAR (I) TO CHAR-KIND (TBLCNT).			43
0089	000890*				
0090	000900	CHAR-COMPARE.			37582
0091	000910	IF CHAR-KIND (J) = CHAR (I)			37582
0092	000920	ADD 1 TO CHAR-COUNT (J)			7445
0093	000930	MOVE 1 TO FOUND.			7445
0094	000940*				
0095	000950	KEKKA-INSATSU.			44
0096	000960	MOVE CHAR-KIND (I) TO LP-CHAR.			44
0097	000970	MOVE CHAR-COUNT (I) TO LP-COUNT.			44
0098	000980	WRITE LP-BUF FROM LP AFTER ADVANCING 1 LINES.			44
0099	000990*				
C100	001000	ATO-SHIMATSU.			1
C101	001010	PERFORM KEKKA-INSATSU VARYING I FROM 1 BY 1			1
C102	001020	UNTIL I > TBLCNT.			
C103	001030	CLOSE SOURCE-FILE PRINT-FILL.			1
C104	001040	STOP RUN.			1

図1 出力結果例

研究開発

- 3) 実行した所とそうでない所が一目瞭然なのでプログラムの検査に有効である。
- 4) 実行回数を手掛りに手続きやデータ構造を変えるなどしてプログラムの効率改善に役立てることができる。
- 5) プログラムが異常終了する時もそれまでに収集された情報を COBOLDAP リストとして出力するのでプログラムのデバッグに有効である。

3. 制限および注意事項

- 1) 入力されるソースプログラムは既に OS IV/F4 COBOL コンパイラにかけて文法違反のないことが確認されているものとする。
- 2) COBOLDAP システムは複数個の OS の COBOL プログラムを処理できるように作成されているため、他の処理系の予約語の一部（特に動詞が多い、表1 参照）も予約語になる。
- 3) COBOLDAP, COBOLDAP - n n n の形の一意名および SYS90n の形の一意名も予約語となる（n n n は3桁の、n は1桁の、数字である）。
- 4) COPY 命令で組込まれる部分は計測の対象とならない。また、その中に STOP RUN が含まれる場合、そこにプログラムの制御が渡ってはならない。
- 5) ファイル定義名として CDPNMSG, CDPNIN, CDPNOUT, CDPNTBL を使用してはならない。これは COBOLDAP システムが作業用に使っているためである。
- 6) 処理できるのは一つの主プログラムだけである。これは複数個のプログラム単位の連続コンパイルができない処理系があるためである。

ACCEPT	EXIT	REWRITE
ADD	GENERATE	SAVED-AREA
ALTER	GO	SEARCH
ASSIGN	GOBACK	SECTION
CALL	IDENTIFICATION	SEEK
CLOSE	IF	SEND
COMMUNICATION	INCLUDE	SET
COMPUTE	INITIATE	SIZE
CONNECT	INVALID	SKIP1
CONSTANT	LINKAGE	SKIP2
COPY	LOCK	SKIP3
DATA	MOVE	SORT
DECLARATIVES	MULTIPLY	START
DISCONNECT	NEXT	STOP
DISPLAY	NOTE	STRING
DIVIDE	OPEN	SUBTRACT
EJECT	OTHERWISE	TREN
ELSE	OVERFLOW	TERMINATE
END	PERFORM	TRANSFORM
ENDCOBOL	PROCEDURE	UNLOCK
END-OF-PAGE	PROCESS	UNSTRING
ENTER	READ	USE
ENTRY	READY	WHEN
ENVIRONMENT	RECEIVE	WORKING-STORAGE
EOP	RELEASE	WRITE
EXAMINE	REPORT	01
EXHIBIT	RESET	1
	RETURN	

表1 COBOLDAP 予約語

4. ジョブ制御文

形式

プロシジャ名	記号パラメータ	プロシジャステップ名
COBOLDAP	$\left[, \text{SYSOUT} = \begin{Bmatrix} \text{A} \\ \text{K} \\ \text{R} \end{Bmatrix} \right]$ (PRVLIB=データセット名)	CDP COB GO

機能

COBOLDAP 前処理部で処理したCOBOLソースプログラムをCOBOLコンパイラで翻訳し、ローダによる結合編集の後実行を行う。

記号パラメータ

SYSOUT 出力クラスを指定する。A (トークン出力), K (カナ付LP), R (端末出力)
 PRVLIB 組込みたい私用ライブラリのデータセット名を指定する。
 関連するDD名 CDP.SYSIN (必須) ソースプログラム用

COBOLコンパイラオプションの標準値

```

APOST NOBATCH NOCHECK NOCLIST COPY NOCSYNTAX NODECK
NODMAP NODYNAM FLAGW NOIDENT NOLIB NOLIL LOCATE NONAME
NONUM OBJECT NOOPT NOPMAP NORENT NOSEQ NOSOURCE NOSTATE
NOSUPMAP NOSXREF NOSYMDMP NOSYNTAX NOTERM NOTEST TRAPC
NOTRUNC VERB NOXREF ZWB SIZE=92160 BUF=16384 LINECNT=60
SPACING=1 SSIN=SYSIN SSOUT=SYSOUT RSV=ALL FILE=STD
NUCLEUS=STD SIGN=STD INTERRUPT=STD RPT=NEW FLOW=NOFLOW
DECNT=NO
  
```

ローダオプションの標準値

```

PRINT NOMAP NOLET CALL NORES NOTERM NODYNAMIC NOALIAS
SIZE=491520 NAME=**GO LINECOUNT=60
  
```

なお、コンパイラオプションを指定する時は必ずNOSOURCE, NOSEQも指定すること。

5. 使用例

以下の例では被計測プログラムはSYSINからデータを入力するものとする。

研究開発

a) ソースプログラム, データともカード入力の場合

```
//      EXEC      COBOLDAP
//CDP.SYSIN      DD      *
//GO.SYSIN       DD      *
//
```

ソースプログラムカード

データ カード

b) データセットからソースプログラムを入力する場合

```
//      EXEC      COBOLDAP
//CDP.SYSIN      DD      DSN=F0039.EXAMPLE.COBOL(SC1),DISP=SHR
//GO.SYSIN       DD      *
//
```

データ カード

c) ソースプログラム, データともデータセットから入力する場合

```
//      EXEC      COBOLDAP
//CDP.SYSIN      DD      DSN=F0039.EXAMPLE.COBOL(SC1),DISP=SHR
//GO.SYSIN       DD      DSN=F0039.EXAMPLE.DATA(DT1),DISP=SHR
//
```

d) 1ジョブ中で複数回使用する例

```
//      EXEC      COBOLDAP
//CDP.SYSIN      DD      DSN=F0039.EXAMPLE.COBOL(SC1),DISP=SHR
//GO.SYSIN       DD      DSN=F0039.EXAMPLE.DATA(DT1),DISP=SHR
//      EXEC      COBOLDAP
//CDP.SYSIN      DD      DSN=F0039.EXAMPLE.COBOL(SC2),DISP=SHR
//GO.SYSIN       DD      DSN=F0039.EXAMPLE.DATA(DT2),DISP=SHR
//
```

e) 私用ライブラリから結合する例

```
// EXEC COBOLDAP, PRVLIB= 'F0039.EXAMPLE.LOAD'
//CDP.SYSIN DD DSN=F0039.EXAMPLE.COBOL(SC1), DISP=SHR
//GO.SYSIN DD DSN=F0039.EXAMPLE.DATA(DT1), DISP=SHR
//
```

f) TSS で利用する例

```
//A003901 JOB F0039, pass word,MSGCLASS=R,MSGLEVEL=(2,0)
// EXEC COBOLDAP, SYSOUT=R
//CDP.SYSIN DD DSN=F0039.EXAMPLE.COBOL(SC1), DISP=SHR
//GO.SYSIN DD DSN=F0039.EXAMPLE.DATA(DT1), DISP=SHR
//
```

この形式のジョブ制御文をSUBMITコマンドでFIBジョブとして実行する。出力はOUTPUTコマンドで出力する。ただし1行80字の端末では見づらいので注意されたい。

6. OSIV/F4 COBOLDAP システムにおけるエラー処理

COBOLDAP システムではプログラム実行中に何らかのエラーが発生するとその時点でプログラムの実行を中止し、それまでに得られた実行回数等の情報を付加したCOBOLDAPリストを出力する。この情報を検討することによってエラーの発生場所をソースプログラム上で容易に推定することができる。OSIV/F4 COBOLDAP システムは実行時に生ずる大部分のエラーを検出してそれに対処することができるが、つぎの異常終了には対処できないので注意されたい。

完了コード	内 容
122	オペレータがDUMPオペランド指定のCANCELコマンドを使用してジョブをキャンセルした時
222	オペレータがCANCELコマンドを使用してジョブをキャンセルした時
322	JOB文またはEXEC文に指定したTIMEパラメータの値より超過してジョブ、ジョブステップまたはプロシジャステップを実行しようとした。
522	ジョブステップ中の全部のタスクが一定時間待ち状態になっていた。
722	SYSOUT DD文のOUTLIMパラメータ値以上にSYSOUTレコードを出力しようとした時
13E	親タスクがDETACHマクロ命令を発行したため、子タスク実行中であつたにもかかわらず強制的に終了させられた。

完了コード	内 容
DQD	ある子タスクの異常終了処理中に再度異常終了が発生したために、ジョブステップ全体の異常終了に切替えた。
1QD	ジョブステップタスクを子タスクに持つステップタスクが異常終了した時、子タスクとして存在するジョブステップタスクがこのコードで異常終了する。

7. システムの構成

COBOLDAP システムの大略について知ることはこのシステムの利用をより円滑にすると考えられるので、システム構成について簡単に述べる。

COBOLDAP システムの構成を図2に示す。このシステムの主要部は前処理部と後処理部で、いずれも COBOL で記述されている。

- ここで、
- NIN : 計測したい COBOL ソースプログラムファイルでカードやデータセット等。(DD名 CDPNIN)
 - NOUT : 前処理部によって計測に必要な文等が挿入された COBOL ソースプログラム用一時的データセット (DD名 CDPNOUT)
 - NTBL : 実行の最後に後処理部が計測結果を出力するために必要な情報を含む一時的データセット (DD名 CDPNTBL)
 - LIBRARY : 収集した情報を実行の最後に編集して出力するリスト編集ルーチン (プログラム名 COBOLDAP) が登録されているデータセット。
 - NMSG : 実行回数等を付加した COBOLDAP リストを出力するリスト用の一時的データセット (DD名 CDPNMSG)

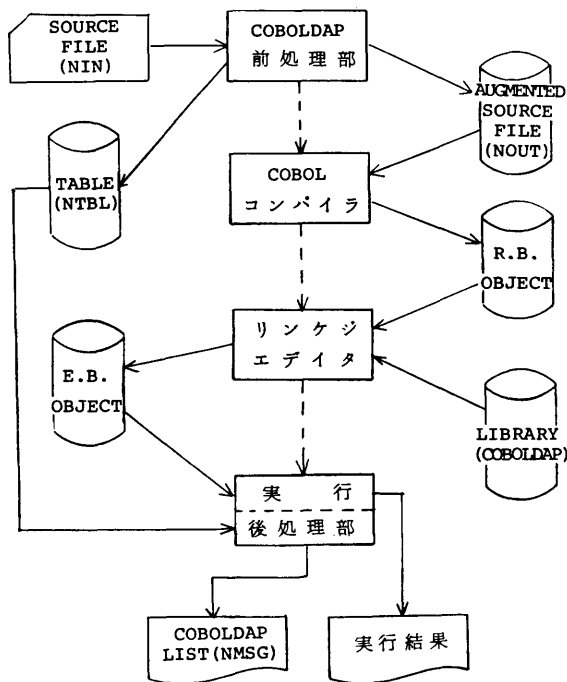


図2 COBOLDAP システム構成図

前処理部によって NIN から NOUT へ編集して出力された被計測プログラムがコンパイル後実行され、計測結果は後処理部によって編集・出力されることになる (図1参照)。

参考文献

- 1) 牛島：`FORDAP-FORTRAN プログラム動的解析システムーについて`，九大大型計算機センター広報，Vol. 9, No. 2, pp. 100-120, 1976
- 2) 牛島，藤村：`M-190 OS IV/F4 FORDAP システム`，九大大型計算機センター広報，Vol. 11, No. 1, pp. 29-33, 1978
- 3) 藤村，牛島：`COBOL プログラム輪郭作成システムの実現と使用について`，九大工学集報，Vol. 50, No. 3, pp. 209-214, 1977
- 4) 藤村，牛島：`COBOL プログラム輪郭作成システムの移し換えについて`，昭和52年度情報処理学会第18回全国大会論文集53, 1977
- 5) 藤村，牛島：`COBOL プログラム輪郭作成システムの使用について`，東大大型計算機センターニュース，Vol. 10, No. 3, pp. 31-37, 1978