

OSIV/F4におけるPL/ I 使用の実際

品川, 一郎
九州大学工学部電気工学科

高木, 成夫
九州大学工学部電気工学科

<https://doi.org/10.15017/1472541>

出版情報 : 九州大学大型計算機センター広報. 11 (3), pp.199-214, 1978-09-01. 九州大学大型計算機センター
バージョン :
権利関係 :



OS IV / F4 における PL / I 使用の実際

品川一郎*・高木成夫*

1. はじめに

OS IV / F4 PL / I は機能の豊富さ、実行速度などあらゆる点で FACOM 230-75 時代の M-VII PL / I に比べてすぐれていると思われるが、PL / I でプログラムを書いて実行しようとするといくつかの問題点が出てくる。その中でもとりわけ切実なのは、バックアップソフトウェアの欠如であろう。FORTRAN を使用する場合、科学技術計算用サブルーチンライブラリ (SSL, SSL II), プロッタサブルーチンパッケージ (PSP), 九大ライブラリ, FORDAP システムなどを容易に利用することができるが、PL / I にはこのようなものはない。また XY プロッタやグラフィックディスプレイを使用する場合は、すべて FORTRAN サブルーチン群 (PSP, GSP (グラフィックサブルーチンパッケージ)) を介在させなければならない。しかしながら OS IV / F4 PL / I には異種言語間結合機能があり、これを利用することにより PL / I から FORTRAN の SSL を呼び出したり、PL / I の結果を XY プロッタに出力したりすることが可能になった。

本稿では、PL / I 固有の便利な機能と、FORTRAN との結合とを中心にして PL / I の実際の使用法をバッチ処理と TSS にわけて紹介する。

2. バッチ処理における PL / I の使用法

2.1 PL / I のカタログドプロシジャ

九大センタには PL / I 関係のカタログドプロシジャ (以下カタプロと略す) として、以下の 4 種類が登録されている。C, CL, CLG, CG 等の意味は FORTRAN のカタプロの場合と同じである。普通ユーザが定義すべきデータセットは、ソースプログラムと実行時データで、この一番基本的な場合のジョブデックの作り方は、「利用の手引き ジョブ制御文編」p.p. 24-27 にあるが、FORTRAN の場合と同様である。

カタプロ名	機 能
PLIXC	PL / I プログラムの翻訳だけ行なう。
PLIXCL	PL / I プログラムの翻訳、結合編集を行なう。
PLIXCLG	PL / I プログラムの翻訳、結合編集、実行を行なう。
PLIXCG	PL / I プログラムの翻訳、ローダによる実行を行なう。

2.2 コンパイラオプション

ここでは PL / I コンパイラの数多いオプションの中から、特にデバッグに関連して便利と思われるものを紹介する。なおオプション名のあとに () でくくってあるのは省略形である。

○ ATTRIBUTE (A)

プログラム中で使われている名標 (変数名、手続き名、ラベル等) の属性を印刷する。このオプションは次に述べる XREF オプションとともに指定すると効果的である。

○ XREF (X)

* 九州大学工学部電気工学科 (プログラム相談員)

名標の相互参照 (cross-reference), すなわちある名標が, どの文で宣言され, どの文で, 値の定義や参照をされているかを, 内部文番号 (ISN) で表示する。

○ NEST

PL/I プログラムはブロック構造をとるため, プログラムによっては内部ブロックや DO グループの入れ子の関係が複雑になる。このオプションを指定すると, ブロックのレベルや, DO グループの入れ子の深さがソースリストの左側に出力される。

○ GOSTMT (GS)

実行時にエラーが発生すると通常はエラーの発生場所が16進のオフセット(相対番地)で出力されるが, この情報からエラーの発生箇所を決定するのは容易でない。しかしこのオプションを指定するとエラーの発生箇所が, ISNで表示されデバッグが容易になる。

○ OPTIMIZE (OPT)

オブジェクトプログラムの最適化のレベルを指定する。このオプションが指定されると, コンパイラは翻訳時に, 実行時の記憶領域が小さくかつ実行速度が速くなるようなオブジェクトを生成する。ただし翻訳に要する時間は増加する。また, あるブロックに対しどの程度まで最適化を許すかは, PROCEDURE文や BEGIN文の ORDERオプションで指定する。また PROCEDUREブロックの関数参照に対してはREDUCIBLEオプションがある。

○ COUNT (CT)

実行時の各文の実行回数を結果の最後に出力する。これは OSIV/F4 FORTRANのダイナミックプロフィールと同じ機能である。OSIV/F4 PL/Iには, FORDAPシステムのような見やすい実行時のプロフィールを出力するシステムがないため, 実行時のプログラムの動的なふるまいは, このCOUNTオプションと後述のFLOWオプションを用いて調べなければならない。ただしこのオプションを指定すると計測のためのオーバーヘッドがかなりあるため²⁾, デバッグが終わったら取りはずしたほうがよい。

○ FLOW

実行時のプログラムの制御の流れを記憶することを指定する。この記憶された制御の流れは SNAPオプション付のオン単位が実行されるごとに, DD名SYS PRINTをもつデータセット (ふつうラインプリンタ) に出力される。オン単位の実行にさかのぼってどのくらい流れを出力するかは, 翻訳時と, さらにプログラム中でも宣言しなければならない。

コンパイラオプションの与え方には次の二つの方法があり, これはFORTRANの場合と同じである。

○ EXEC文のパラメータとして与える。

○ 各外部手続きの前に PROCESS文をそう入し, そこでオプションを指定する。

オプションの採用は次のような規則に従って行なわれる。

1) どのような場合にこのオプションを指定すればよいかについては, 広報 Vol. 11 の No. 2 p. 80 が参考になるであろう。

2) 筆者が使用した例ではこのCOUNTオプションを指定したため実行時間が約2倍になった。

- ① EXEC文のオプションを左から順次採用。
- ② PROCESS文のオプションを左から順次採用。
- ③ 指定されていないオプションに対しては省略時解釈を適用する。省略時解釈はシステム編集時に決定される。

なお先に指定されたオプションと矛盾するオプションが指定された場合新しいオプションに従う。
コンパイラオプションの詳細と最適化については、使用手引書を参照されたい。¹⁾

次に例題をいくつか示す.

例 2.1 この例では，EXEC文とPROCESS文でオプションを与えてある。

```
/// EXEC PLIXC, PARM. PLI=▼A,X,NEST▼
```

```

//PLI.SYSIN DD *

```

```
*  PROCESS  NA,NX  ;
```

メイン プログラム

* PROCESS NONEST :

外部手続表

/ *

このようにオプションを指定すると、メインプログラムに対してはNESTと、AおよびXの否定形のオプションが採用される。外部手続きに対しては、A、X、およびNESTの否定形のNONESTが採用される。他の指定されないオプションに対しては省略時解釈が適用される。

なお、コンパイラオプションの具体的な例は、文献(1)に詳しい。

2.3 異種言語間結合機能²⁾

OSIV/F4 PL/I コンパイラの異種言語間結合機能は、PL/I と、アセンブラ、FORTRAN、COBOL との相互呼び出しを可能にしている。この節では、PL/I から FORTRAN プログラムを呼び出す場合について説明する。

○ プログラミング上の注意

j) 呼び出される FORTRAN 副プログラム名は、PL/I プログラム中で、ENTRY 属性を持たせ、外部手続きであることを宣言し、必要なパラメータの属性を宣言しなければならない。

ENTRY 属性の宣言の一般形

DCL 外部入口定数 ENTRY (パラメタ記述子, ……)

OPTIONS (FORTRAN) ;

外部入口定数 この場合呼び出される FORTRAN 副プログラム名のこと

パラメタ記述子 アーギュメント（引数）によってデータの受け渡しを行なう場合、各パラメタの属性を記述する。アーギュメントとパラメタのそれぞれの属性が一致し

- 1) FACOM OS IV/F4 PL/I使用手引書(以下使用手引書と略す)

コンパイラオプションについては P.P.3 - 23

最適化については第8章に詳しい。

- 2) OS IV/F 4 PL/I 使用手引書 p.p.93-100

ている場合は省略可能であり、その場合は、* を書く。

また、FORTRANとPL/Iでは集合体(配列)の添字の配置が異なるが、この問題を解決するため異種言語間結合機能によりダミーアーギュメントが作成される。このダミーアーギュメント作成を制限したい場合には、ENTRY属性の宣言の際にオプションNOMAP等を使用する。

ii) データの受け渡しの方法には次の三つの方法がある。

- (1) PL/IとFORTRANがデータセットを共用する。
- (2) アーギュメントの形式でデータの受け渡しを行なう。
- (3) 静的記憶域の形式でデータの受け渡しを行なう。静的記憶域とは、PL/IではEXTERNAL STATIC 属性をもつ記憶域クラス、FORTRANではCOMMONブロックのことである。

どの方法を採用するかは、プログラマが判断すべき問題であるが、SSLやPSPのようなライブラリサブルーチンを呼び出す場合は、データの受け渡しはアーギュメントによることになっているので、ここでは(2)の形式についてのみ説明する。ただしどの方法にも共通して注意すべきことは、FORTRANとPL/Iの間で受け渡しをされるデータは、必ず属性(型)が合っていないなければならないということである。PL/IとFORTRANのそれぞれのデータの大部分には、直接の対応関係があるが、注意すべき点が二つある。その一つは、FORTRANの整数型は4バイトであり、PL/Iのそれ(BIN FIXED(15,0))は2バイトであるから、PL/IからFORTRANのSSLや、PSPを呼ぶ時はFORTRANにあわせて、PL/Iの側でも、引数として使われるものに対しては4バイト(BIN FIXED(31,0))の宣言をしておく必要がある。もう一つは、PSPのAXISサブルーチンのように、文字を引数とする場合である。PL/Iには文字型があるが、現在のFORTRANには、文字型という型がなく、したがって引数の型を合わせることができず、強引に文字を引き渡そうとするとコンパイラエラーとなり実行は保証されない。しかし筆者がテストした結果では、後述の例題のような方法を使えば、正しく実行してくれる。

1)

例 2.2 富士通作成 FORTRAN SSL II EIG1 を PL/I から呼び出す。

EIG1 は実行列 A の固有値と正規化固有ベクトルを計算する FORTRAN 副プログラムである。下に示した PL/I プログラムは行列 A を読み込んでその固有値、固有ベクトルを FORTRAN 副プログラム EIG1 で計算し、EIG1 からのリターンコード(ICON)が2000以上なら ICON の値を印刷して止まり、そうでない時は、さらに処理を続ける。(以後の処理のプログラムは省略してある。)プログラムについて説明すると、行番号20と50の文で、N, K, ICON は、FORTRAN と型を合わせるために、4バイトの宣言をしている。行番号30の文では、EIG1 が FORTRAN の外部手続きであることを宣言し、パラメタ記述子も記述している。その際第2、第3、第4パラメタ以外は、アーギュメントとパラメタの属性が一致するので、*を記入している。

1) 本サブルーチンの使用法については、FACOM FORTRAN SSL II 使用手引書 p.p.57-60

```

00010  TESTSSL :  PROC  OPTIONS(MAIN)  ;
00020      DCL ( N, K ) BIN FIXED(31,0) ,
00030          EIG1 ENTRY(*,BIN FIXED(31),BIN FIXED(31),BIN FIXED(31),
00040                      *,*,*,*,*)  OPTIONS(FORTRAN) ,
00050          ICON BIN FIXED(31)  ;
00060
00070      GET LIST(N)  ;    K = N  ;
00080
00090      BEGIN  ;
00100          DCL  A(N,N), ER(N), EI(N), EV(K,N), VW(N)  ;
00110          GET LIST(A)  ;
00120          PUT EDIT( ' INPUT ARRAY < A >' ,A)
00130              ( LINE(2),COL(5),A,SKIP(2),(N)(SKIP,COL(5)
00140                  ,(N)(E(11,3)))));
00150      CALL EIG1(A,K,K,0,ER,EI,EV,VW,ICON)  ;
00160
001701)      IF  ICON ^ = 0
00180          THEN DO  ;
00190          PUT DATA(ICON)  ;
00200          IF  ICON >= 20000
00210              THEN STOP  ;
00220          END  /* OF DO GROUP  */  ;

```

.

.

.

.

例 2.3 PL/I から PSP を呼び出す。

PSP を呼び出す場合のプログラム上の注意事項は前の例の SSL を呼び出す場合と同じである。ただ AXIS に対して文字列を引き渡す場合は、PL/I の文字変数として、引き渡す。実験してみたところ、文字定数を引き渡すと文字が正しく出力されない。また、可変長の文字列 (VARYING 属性をもつ文字列) として引き渡しても正しく出力されない。文字列を引き渡す場合翻訳時に、このアーギュメントは FORTRAN に対応する属性がないとのメッセージが出て、E レベルのエラーとなるが、上に述べたように固定長の文字変数として渡す限り、結果は正常である。以下に PL/I から PSP を呼び出すプログラムの一部を掲げる。

1) 記号 “^” は TSS 端末を使用するとき、“¬” (否定記号) の代用として用いる。また、F1510 端末の場合、“|” (OR 記号) の代用としては “!” を用いる。TTY 手順端末では “|” は、“|” または “;” である。

```

00420      DCL
00430          (( PLOTS, PLOT) ENTRY(FLOAT,FLOAT, BIN FIXED(31,0)),
00440              PLOTE ENTRY(*),
00450              SCALE ENTRY(*,FLOAT, BIN FIXED(31,0),BIN FIXED(31,0)),
00460              LINE  ENTRY(*,*,BIN FIXED(31,0),BIN FIXED(31,0),
00470                          BIN FIXED(31,0),BIN FIXED(31,0)),
00480              AXIS  ENTRY(FLOAT,FLOAT,*,BIN FIXED(31,0),
00490                          FLOAT,FLOAT,*,*) )
00500      OPTIONS(FORTRAN) ;
    {
04590      DCL ( CH_STEP CHAR(4)  INIT('STEP'),
04600          CH_CTL  CHAR(9)  INIT('CONTROL_U'),
04610          CH_TRJ  CHAR(10) INIT('TRAJECTORY')) ALIGNED ;
04820      CALL PLOT(0.0,-11.0,-3) ;
04830      CALL LINE(STEP_U,U_SEQ,20,1,0,0) ;
04840      CALL AXIS(0.0,0.0,CH_STEP,-4,13.0,0.0,
04850                  STEP_U(21),STEP_U(22)) ;
04860      CALL AXIS(0.0,0.0,CH_CTL,9,7.0,90.0,
04870                  U_SEQ(20),U_SEQ(21)) ;

```

○ ジョブ制御文の作り方

PL/I から FORTRAN の SSL や PSP を呼ぶ際の制御文の作り方の基本は、PL/I のレジデントライブラリを定義する DD 名 SYSLIB に SSL や PSP のはいたデータセットを連結することである。この具体的な方法としては、何通りかが考えられるが、一番簡単と思われるものを次にあげる。

例 2.4

- ① // EXEC PLIXC
 - ② // PLI.SYSIN DD *
- PL/I ソースプログラム
- ③ // EXEC LKED, PRVLIB=▼F1234.LM.LOAD▼
 - ④ // LKED.SYSLIN DD DSN=&&OM, DISP=(OLD,DELETE)
 - ⑤ // EXEC GO,LOADDS=▼&&LM▼, COND=(9,LT)
 - ⑥ // GO.SYSIN DD *
- PL/I 実行時のデータ
- ⑦ // GO.FT16F001 DD SYSOUT=(Q,XYPLOT)
 - ⑧ // GO.SYSGO DD *

FORTRANで機番5からの入力データ

①でまず PL/I ソースプログラムの翻訳を行なう。②はソースプログラムを定義する DD 文である。次に③で結合編集を行なう。このステップの実行により、SSL や PSP などはすべて組込まれる。私用ライブラリも組み込む時は③で例のようにする。④はオブジェクトモジュールを定義する DD 文で必ずこの位置におく。⑤は実行を指示する。⑥は PL/I 実行時に、PL/I のシステム標準入力ファイルから入力するデータセットを定義している。⑦は、XYプロッタを使う場合に必ず定義する。⑧は、もし PL/I から呼ばれた FORTRAN プログラムが、機番 5 から入力する場合にその入力データを定義する DD 文である。⑥や⑧は、データがない場合にはもちろん不要である。ソースプログラムやデータをデータセットから読む場合は、通常の場合と同様である。

3. TSS下における PL/I の使用法

3.1 データセットの作成

PL/I の場合も、他の言語と同様に TSS 下で EDIT コマンドを用いて、データセットの作成及び更新ができる。ここでは、まずソースデータセットのレコード形式について述べ、次に EDIT コマンドを用いてソースデータセットを作成するための方法について PL/I システムとの関係で注意すべき点を中心に説明する。

i) ソースデータセットのレコード形式

PL/I ソースデータセットのレコード形式としては次の 2 通りがある。

○ 固定長ブロック化 (FB)

論理レコード長 : 100 バイト以下

ブロック長 : 論理レコード長の整数倍で 400 バイト以下

この形式の利点は空白行が認められることと、論理レコード長を 80 とすれば、カードとの互換性があることである。

○ 可変長ブロック化 (VB)

論理レコード長 : 104 バイト

ブロック長 : 400 バイト以下

この形式では空白行が認められないが、固定長ブロック化形式よりも領域が少なくすむ。

ii) EDIT コマンド

EDIT コマンドにはデータセットの性質を与えるために多数のオペランドが用意されているが、ここでは、PL/I と関係する部分の構文と内容を示す。

構文 : EDIT データセット名 { OLD | NEW [BLOCK (n₁)]
[LINE (n₂)] }

PLI [([m₁ [, m₂]]

[CHAR60 | CHAR48])]

[SCAN | NOSCAN] [NUM | NONUM]

下線部は省略時解釈である。

一般則 :

(a) データセット名

TSS の下で使用するデータセットの名前を与える。これは TSS で定められたデータセット命名規約に従う。

(b) { OLD | NEW [BLOCK (n_1)] [LINE (n_2)] }

データセットが既に作成されたものであるか 新たに作成するものであるかを指定する。OLD は既存のものを示し、NEW は新たに作成するものであることを示す。

BLOCK 及び LINE は指定されたデータセットのブロック長及びレコード長を与えるものである。新たにデータセットを作成する時に LINE を指定しないと可変長ブロック化形式となる。 n_1, n_2 は符号のない整数であって表 3.1 に示す制限を満足しなければならない。

(c) PLI ([m_1 [, m_2]])

データセットが PLI ソースプログラムであることを示す内容識別修飾子である。 m_1, m_2 は符号のない整数であって、各行の m_1 から m_2 までの欄が PL/I のソースプログラムであることを示す。(ソースマージン)

m_1 及び m_2 の省略時の値は、固定長ブロック化形式で論理レコード長が 80 のときは、2 と 72、可変長ブロック化形式のときは NONUM が指定されると 2 と 100、NUM 指定であると 10 と 100 がとられる。

(d) { CHAR60 | CHAR48 }

PL/I ソースプログラムが 60 字セットで書かれているか、48 字セットで書かれているかを示す。九大センタに接続されている端末は、否定記号と論理和記号を、それぞれ “\ ” と “! ” で置きかえることにより、60 字セットで使用可能である。

(e) { SCAN | NOSCAN }

INPUT モードで入力したテキストの各行を、文単位に構文チェックするかどうかを指定する。

(f) { NUM | NONUM }

データセットの各行に行番号を付けるかどうかを指定する。NUM を指定したときの行番号の位置は表 3.1 に示したとおりである。

表 3.1

	LINE (n_2)		BLOCK (n_1)		レコード 形 式	NUM が指 定された ときの NUM 位置
	省略時の値	制 限 値	省略時の値	制 限 値		
P L I	—	$n_2 \leq 100$	400	$n_1 \leq 400$	F B	終り 8 文字
	104	—	400	$n_1 \leq 400$	V B	始め 8 文字

例 3.1 データセット名 SC

レコード形式 : 固定長ブロック化, 論理レコード長 80 バイト

ブロック長 400 バイト

ソースマージン(2, 72)のソースデータセットを新規作成する。

```

READY
EDIT SC NEW LINE(80) PLI(2,72)
INPUT
00010  PLITSS1) : PROC OPTIONS(MAIN) ;
00020          DCL (A,B,C)(3,3),WK(3) ;
00030          GET LIST(A,B) ;
00040          DO I = 1 TO 3 ;
00050          DO J = 1 TO 3 ;
00060          WK = A(I,*)*B(*,J) ;
00070          C(I,J) = SUM(WK) ;
00080          END ; END ;
00090          PUT EDIT(C)(SKIP,(3)E(15,7)) ;
00100          END PLITSS ;
00110 (復改)
EDIT
END S
READY

```

なお、下線をほどこした部分はユーザが打鍵したことを示す。

3.2 TSSにおける翻訳と実行

端末から PL/I コンパイラを用いて翻訳と実行を行う方法は表 3.2 に示されるように、3 通りある。ここでは 3 通りの手順を例題を示しながら説明する。

なお、SSL, SSL II, 私用ライブラリ, PSP を含むプログラムの実行に関しては 3.3 でくわしく述べる。

1) このプログラムは行列 A と行列 B の積を求めるものである。

表 3 . 2

1)

手 順	使用コマンド	説 明
1	P L I	PL/I ソースプログラムを入力し、翻訳を行なう
	LOADGO	オブジェクトモジュールを入力し、ローダによる実行を行なう。
2	P L I	1に同じ
	L I N K	オブジェクトモジュールを入力し、ロードモジュールを作成する。
	C A L L	ロードモジュールを実行する。
3	R U N又は R U Nサブ コマンド	翻訳から実行までを連続して行なう。処理の形態としては、上記の 手順1と同じである。

i) P L I , L O A D G O

例 3.2 例 3.1 で作成したソースプログラムを実行させる。このときコンパイラオプションとして、OPT (最適化) と LMSG (長文メッセージ) を指定する。

ソースプログラム S C . P L I
オブジェクトモジュール S C . O B J

READY

PLI SC.PLI OPT LMSG

JMG1041I PL/I COMPILER INVOKED

FACOM OSIV/F4 PL/I COMPILER V01L18 DATE 78.06.29 TIME 15.08.14

** OPTIONS SPECIFIED **

OPT(2), LMSG

** COMPILER DIAGNOSTIC MESSAGES **

MSG-ID L NUMBER MESSAGES

JMG890I-I 10 'ORDER' OPTION IS APPLIED. GLOBAL OPTIMIZATION IS
RESTRICTED FOR THIS BLOCK.

** END OF COMPILER DIAGNOSTIC MESSAGES **

STATICS: CPU TIME 0.00 MINUTES, SPILL RECORDS 0, PAGE SIZE 4051,
SOURCE STATEMENTS 10, OBJECT SIZE 1234, HIGHEST SEVERITY CODE 00
READY

LOADGO SC.OBJ PLIBASE

1) TSSのコマンドのくわしい説明はFACOM OSIV/F4 TSSコマンド文法書を参照されたい。

```

1)
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0
2)
+: 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0

```

```

1.0000000E+00 2.0000000E+00 3.0000000E+00
4.0000000E+00 5.0000000E+00 6.0000000E+00
7.0000000E+00 8.0000000E+00 9.0000000E+00

```

READY

ii) PLI, LINK, CALL

例 3.3 例 3.1 で作成したプログラムを実行させる。

```

ソースプログラム      SC.PLI
オブジェクトモジュール SC.OBJ
ロードモジュール      SC.LOAD(MEM1)

```

READY

PLI SC.PLI OPT LMSG

JMG1041I PL/I COMPILER INVOKED

FACOM OSIV/F4 PL/I COMPILER V01L18 DATE 78.06.29 TIME 15.17.55

** OPTIONS SPECIFIED **

OPT(2), LMSG

** COMPILER DIAGNOSTIC MESSAGES **

MSG-ID L NUMBER MESSAGES

JMG8901-I 10 'ORDER' OPTION IS APPLIED. GLOBAL OPTIMIZATION IS,
RESTRICTED FOR THIS BLOCK.

** END OF COMPILER DIAGNOSTIC MESSAGES **

STATICS: CPU TIME 0.01 MINUTES, SPILL RECORDS 0, PAGE SIZE 4051,
SOURCE STATEMENTS 10, OBJECT SIZE 1234, HIGHEST SEVERITY CODE 00

READY

LINK SC.OBJ LOAD(SC(MEM1)) PLIBASE

MEMBER NAME MEM1 NOW ADDED TO DATA SET.

READY

CALL SC(MEM1)

:

1) : は入力データ促進文字である。

2) + : は1回のGET文の実行でさらにデータを要求していることを示す。

```

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0
+ : 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0

```

```

1.0000000E+00 2.0000000E+00 3.0000000E+00
4.0000000E+00 5.0000000E+00 6.0000000E+00
7.0000000E+00 8.0000000E+00 9.0000000E+00

```

READY

iii) RUN

RUNコマンド又は、EDITのRUNサブコマンドを用いると、翻訳から実行までを連続して行うことができる。しかし、コンパイラオプションとしては、ソースマージン、文字セット、長文メッセージの指定しかできない。

例 3.4 例 3.1 で作成したプログラムを実行させる。

ソースプログラム SC.PLI

READY

RUN SC.PLI

JMG1041I PL/I COMPILER INVOKED

FACOM OSIV/F4 PL/I COMPILER V01L18 DATE 78.06.29 TIME 15.21.35

**** OPTIONS SPECIFIED ****

OBJ

**** NO COMPILER DIAGNOSTIC MESSAGES ****

STATISTICS: CPU TIME 0.00 MINUTES, SPILL RECORDS 0, PAGE SIZE 4051,
SOURCE STATEMENTS 10, OBJECT SIZE 1190, HIGHEST SEVERITY CODE 00

:

```

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0
+ : 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0

```

```

1.0000000E+00 2.0000000E+00 3.0000000E+00
4.0000000E+00 5.0000000E+00 6.0000000E+00
7.0000000E+00 8.0000000E+00 9.0000000E+00

```

READY

3.3 TSS下におけるFORTRANとの結合

プログラミング上注意すべき点はバッチ処理の場合と全く同じであるが、TSSではデータセットの割り当てをDD文ではなくALLOCATEコマンドで行なう。TSS下でPL/Iを実行する際FORTRANのSSLや私用ライブラリを呼ぶ時は、これらのロードモジュールのはいったデータセ

ットを一次ライブラリとして、RUNコマンドやLOADGOコマンドのLIBオペランドで指定するか、あるいはALLOCATEコマンドでDD名を指定し、二次ライブラリとしてRUNコマンドやLOADGOコマンドのLIBDDオペランドで指定する。この場合のDD名は、すでに使用されている名前であってはならない。それを確認するには、LISTALCコマンドをSTATUSオペランドを指定して打ち込めばよい。

例3.5 PL/Iからすでに作成されたFORTRAN私用ライブラリ中のサブルーチンSUBを呼び出し実行する。私用ライブラリのデータセット名はFSLIB.LOADでメンバ名はSUBである。

PL/Iから呼ばれるFORTRANサブルーチン

```
00010 SUBROUTINE SUB(A)
00020 REAL*4 A(3,2),B(3,2)/1.,4.,2.,5.,3.,6./
00030 DO 1 I=1,3
00040 DO 1 J=1,2
00050 A(I,J)=A(I,J)+B(I,J)
00060 1 CONTINUE
00070 RETURN
00080 END
```

PL/I プログラム データセット名 P.PLI

```
00010 TEST : PROC OPTIONS(MAIN) ;
00020 DCL A(3,2) INIT(1.,2.,3.,4.,5.,6.),
00030 SUB ENTRY(*) OPTIONS(FORTRAN) ;
00040 PUT EDIT('A(BEFORE) =',A)(R(FMT)) ;
00050 CALL SUB(A) ;
00060 PUT EDIT('A(AFTER) =',A)(R(FMT)) ;
00070 FMT : FORMAT(SKIP(3),A(11),(3)(COL(14),(2)(E(10,2)))) ;
00080 END TEST ;
```

ソースプログラムの翻訳

READY

PLI P NOP

PL/I COMPILER INVOKED

FACOM OSIV/F4 PL/I COMPILER V01L18 DATE 78.06.29 TIME 16.19.02

** OPTIONS SPECIFIED **

NOP

** COMPILER DIAGNOSTIC MESSAGES **

MSG-ID L NUMBER MESSAGES

JMG770I-I 50 DUMMY ARGUMENT IS CREATED FOR ARGUMENT 1 TO 'SUB'.

** END OF COMPILER DIAGNOSTIC MESSAGES **

STATISTICS: CPU TIME 0.00 MINUTES, SPILL RECORDS 0, PAGE SIZE 4051,
SOURCE STATEMENTS 8, OBJECT SIZE 1858, HIGHEST SEVERITY CODE 00

PL/I と FORTRAN のプログラムはそれぞれ先に示した通りである。このプログラムは PL/I で定義された配列 A と FORTRAN で定義された配列 B を加えるだけである。PL/I と FORTRAN では配列の添字の並び方が異なるため、¹⁾ 翻訳時にダミーアークメントが作成されている。このプログラムを LOADGO コマンドで実行した結果を次に示す。

実行に先立って、FORTRAN ライブラリのはいったデータセット、SYS1.FORTLIB と、私用ライブラリ FSLIB.LOAD を連結して、DD 名 GLIB を与え、LOADGO コマンドの LIBDD オペランドで二次ライブラリとして指定してある。結果を見ると、配列の対応関係はちゃんとついていることがわかる。

A (1 , 1)	A (1 , 1)
A (1 , 2)	A (2 , 1)
A (2 , 1)	A (3 , 1)
A (2 , 2)	A (1 , 2)
A (3 , 1)	A (2 , 2)
A (3 , 2)	A (3 , 2)

PL/I

FORTRAN

配列の記憶場所の対応

READY

ALLOC DA('SYS1.FORTLIB' FSLIB.LOAD) F(GLIB) SHR

READY

LOADGO P.OBJ LIBDD(GLIB) PLIBASE

A(BEFORE) = 1.00E+00 2.00E+00
 3.00E+00 4.00E+00
 5.00E+00 6.00E+00

B = $\begin{bmatrix} 1 & 5 \\ 4 & 3 \\ 2 & 6 \end{bmatrix}$

A(AFTER) = 2.00E+00 7.00E+00
 7.00E+00 7.00E+00
 7.00E+00 1.20E+01

4. その他

4.1 PL/I 私用ライブラリの作成

PL/I はブロック構造をとる言語であり手続きをすべて内部的にしてメインプログラム中に納めしてしまうこともできるが、一部の手続きを外部手続きとし、さらにそれをロードモジュールに落とし、私用ライブラリとして使うこともできる。PL/I の手続きを外部的にするには、翻訳時に各手続きの前に PROCESS 文をそう入する。また他の外部手続きからこの外部手続きを引用する場合には、

1) FORTRAN では左側の添字が変化する順に記憶場所が割り付けられるが、PL/I では右側からである。

引用される外部手続きに対し、ENTRY属性を宣言しなければならない。その方法は、2.3で述べたのと同様である。(OPTIONS(FORTRAN)はもちろん不要である。)私用ライブラリを作成する場合には、さらにコンパイラオプションとして、NAMEオプションを指定する。その他の点はFORTRANの私用ライブラリの作成法と同様である。¹⁾次にジョブ制御文の例を示す。

例 4.1

```
// EXEC PLIXCL, PARM.PLI=NAME,
//          PARM.LKED=NCAL
// PLI.SYSIN DD *
* PROCESS   ;
    外部手続き 1
* PROCESS   ;
    外部手続き 2
// LKED.SYSLMOD DD DSN=F1234.LM.LOAD,
//          DISP=(NEW,CATLG),UNIT=PUB,
//          SPACE=(TRK,(10,5,3),RLSE)
```

また、TSSにおいてもPLIコマンドとLINKコマンドを用いて同様に私用ライブラリを作成することができる。²⁾

4.2 実行時の入出力について

システム標準入出力ファイルSYSIN, SYSPRINT以外から入出力を行なおうとする場合には、プログラム中でファイルを宣言し、さらにこれをジョブ制御文のDD文で定義すればよい。PL/Iは、豊富な入出力機能を持つが、ここでは実行時にラインプリンタ(LP)に出力させる場合だけを述べる。

システム標準出力ファイルは、ファイル名SYSPRINT, DD名SYSPRINTを持ち普通はLPに割り当てられる。プログラムで何も指定しなければ、LPに出力する時には、1ページに60行(PAGESIZE(60))、1行に120文字(LINESIZE(120))印刷するようになっている。このSIZEを変更したい場合にはOPEN文のオプションとして指定すればよい。また、LP出力ファイルを区別したい場合に、プログラム中で新たにファイルを宣言してFILE, PRINT両属性を与え、ジョブ制御文に対応するDD文を追加すればよい。なおPRINT属性をもつファイルは、PAGESIZE(60)、LINESIZE(120)が省略値としてとられる。次にこの具体的な例を示す。

例 4.2

```
① DCL  SYSPOT FILE PRINT ;
    :
```

1) 利用の手引き データセット編 p.p. 7-8

2) OS IV/F4 PL/I使用手引書 第4章～第6章


```

② OPEN FILE(SYSPOT)    PAGESIZE(66) ,
    FILE(SYSPRINT) LINESIZE(135) ;
    :
③ PUT FILE(SYSPOT) EDIT(.....)(.....) ;

```

このプログラムはごく一部の抜粋である。①ではPRINT属性をもつSYSPOTというファイルを定義している。②で2つのファイルを開き、その時にPAGESIZEとLINESIZEを指定している。SYSPRINTはPAGESIZE(60), LINESIZE(135), SYSPOTはPAGESIZE(66), LINESIZE(120)となる。③は具体的にSYSPOTにデータを書き出す文である。

対応するDD文の与え方は、カタプロPLIXCLGを使った場合次のようにする。

なおこのようにLP出力ファイルを
区別すると、出力検索システムを利用
する場合データセット単位の削除がや
りやすくなり効果的と思われる。

```

// EXEC PLIXCLG
// PLI.SYSIN DD *

```

ソースプログラム

5. おわりに

OS IV/F4 PL/I の使用に際し、
知っていると便利と思われる点をいくつ
か紹介した。筆者らの経験もまだ不十分
ではあるが、この小文がPL/Iを使わ
れる方の参考になれば幸いである。なお

```

// GO.SYSIN DD *

```

実行時データ

```

// GO.SYSPOT DD SYSOUT=A,
//      DCB=RECFM=UA

```

本稿をまとめるに当たっては、文献(1)を大いに参考にさせていただいた。おわりに有益な助言をいただいた九州大学工学部情報工学科牛島教授に深く感謝の意を表する。

参 考 文 献

- 1) 黒島 : "PL/Iについて(1),(2)" 京大センタ広報, VOL.10, No.4~5
- 2) 富士通: FACOM OSN PL/I 文法書 64SP-3060-1
- 3) 富士通: FACOM OSN/F4 PL/I 使用手引書 64SP-3070-2