

Title	FORTRANとの比較によるPL/I 入門 (1)
Author(s)	塩野, 充
Citation	大阪大学大型計算機センターニュース. 1980, 36, p. 63-71
Version Type	VoR
URL	https://hdl.handle.net/11094/65439
rights	
Note	

Osaka University Knowledge Archive : OUKA

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

Osaka University

FORTRANとの比較によるPL / I入門(1)

大阪大学工学部 塩野 充

第1章 雑 論

ACOS(エイコス)という言葉は今や阪大の理工系の学部生、大学院生、教職員の間ではほぼ知らない人がいないくらい浸透したと考えてよいだろう。もし、まだ知らなくて「エイコスで、なんや?」という人がいればその人は余程冗談の好きな人か、或いはかなりエキセントリックな人であろう。しかし、ACOSという言葉が一体どういう略語なのか、という事まで知っている人は案外少ないようである。辞書など引いても勿論載っていない。これは日本電気の広告によれば、4通りぐらいの意味があって、All-round Customer Oriented System, Advanced Concept Oriented System, Advanced COmputer System, Advanced Comprehensive Operating Systemとなっている。老婆心乍ら、英語に弱い人のために、英語に弱い筆者が辞書を引くと、多方面の顧客を指向したシステム、進歩した概念を指向したシステム、進歩した電算機システム、進歩した広汎なオペレーティングシステム、等となる。数年前、国産コンピュータはバラバラでは海外製に対抗できないため3つのグループにまとめられた。その各々にシリーズ名が付けられ、日本電気(NEAC)と東芝(TOSBAC)によるACOSシリーズ、富士通(FACOM)と日立(HITAC)によるMシリーズ、三菱(MELCOM)と沖電気(OKITAC)によるCOSMOシリーズとなった。このうち旧七帝大の全国共同利用大型計算機センターに入っているのはACOSシリーズとMシリーズのいずれも超大型機である。現在、ACOSシリーズは阪大と東北大、Mシリーズは京大と九大に入っている。その他の所はまだ旧タイプであるが、いずれ新シリーズの超大型機と置き換えられるであろう。ところで余談にはなるが、超大型機というのは別名「金喰い虫」とも言われ、開発に莫大な費用がかかり、しかも製品としては決して飛ぶように売れるシロモノではなく、いかなれば9月まで売れ残ったクーラーよりも売れにくいものである。一般の会社が買うコンピュータは小型から中型、ないしせいぜい大型であり、超大型というのは売るといふよりもむしろそのメーカーの技術力を誇示し下位機種イメージアップを図るためのシンボルのような役目を担っているのであろう。乗用車にたとえるならばトヨタのセンチュリー、日産のプレジデントに相当するだろう。こういう製品をプレスティジモデル(Prestige Model)と呼ぶこともある。

話が脱線しすぎたが(古い文章をひもとくと、こういう箇所では「閑話休題」という言葉がきまり文句になっているが、ここではコンピュータという最新テクノロジーの話なので、そういうカビ

くさい言葉は使わないでおこう)、阪大の大型計算機センターにACOSシリーズが初めて入ったのは1977年で、その時入ったのは上から3番目の機種で、ACOSシリーズ77NEACシステム700であった。以下、長いので単にACOS700と書く。その頃はまだ旧型のNEACシリーズ2200モデル700というのが主流で、ACOS700は試験的に入れられており、使う人も少なかった。当計算機センターが本格的にACOS時代に入ったと言えるのは1978年春に上記2機種が撤廃されてACOS800だけになった時であろう。その後、1978年末にACOS900になり現在に至っている。NEACシリーズ2200モデル700の時代からACOS時代に入って変わった事を挙げれば次のような点であろう。まず、誰でも気が付く事から書くと、

1. カードリーダーその他の機器が黒を基調としたデザインになった。
2. ラインプリンタに自動カッターが付いた。
3. ラインプリンタの印字が小さくなった。これは多分印字速度を上げるためであろうが、一つ苦情を述べると英字のオーと数字のゼロの区別が非常につきにくい。目をこらして見るとオーの方がややスリムなのだが、もっと明確に区別できるような字形にしてもらいたい。ユーザの中には目の衰えた年配者もいるからである。それともう一つ、これは標題のPL/Iにも関連するがマイナスとアンダーラインの区別もつきにくい。PL/Iで使用できるアンダーライン(下線)はFORTRANでは使用できないのでなじみが薄いかもしれないが、パンチ機のキーではWのソフト位置にある記号である。これはパンチ機のカード上への印字ではマイナスとは長さも位置も異なるので、明確に区別できるが、ラインプリンタの印字では長さも似ており、位置によってしか識別できない。(尤も、虫めがねで見れば多少長さは違うのだろうが、易者のようにいつも虫めがねを持ってセンターに行くわけにもいかない)。これが、ラインプリンタの調子によって印字の上下位置が乱れると、アンダーラインがマイナスと同じ高さになったりして全く見分けがつかなくなる事がある。アンダーラインの両端に小さくカギを付けるなどして、マイナスと区別し易くしてもらえたらと思う。
4. JCL(ジョブ制御言語)が全面的に変ってしまった。これはユーザにとって迷惑この上ない事であった。それ迄馴れ親しんできたNEACシリーズ2200のJCLが一夜にしてパーとなり、JCLの再勉強を強いられた。若い者は頭も柔軟なのですぐ切り換えができれば、年配の方の中にはかなり難渋された方もいたようである。NEACシリーズからACOSシリーズへの転換というのは、いわば第3世代から第4世代への世代交代であり、一つの大きな過渡期にあたるのでJCLの全面変更ぐらいはやむを得ない事なのかも知れないが、今後はこのような事はしないでほしいと思う。ところで筆者はいつも思うのだが、JCLもFORTRANのようにJISで規格化して全メーカー共通にすればどんなに便利だろうと思う。夢のような話だが。
5. その他、処理時間が速くなったとか、FORTRANのJIS以外の部分が少し変わったとか、

いろいろあろう。

以上列挙した他に一つ、非常に大きな変化があった。これには気付かない人が多いか、或いは気付いていても気にとめない人が多いのではないかと思うが、PL/I（ピーエルワン）なる言語が使用可能になったのである。

第2章 PL/Iのすすめ

随分前置きが長かったが、この辺からいよいよ標題の入口へ入る。NEACシリーズ2200の頃にはPL/Iは使えず、わずかにBPLという、PL/Iのサブセット的な言語が入っていたが、本格的なPL/Iというには程遠いものであった。京大の大型計算機センターでもFACOM230-75の時代、PL/Iはあったが、M190になってから複素数も扱える本格的なPL/Iになった。当計算機センターもACOS時代になって本格的なPL/Iが入ったといえよう。従って、ここ2～3年でPL/Iは大学の計算機センターで非常に普及した、といえる筈なのである。ところがところが、それは形式的な面ではそうであっても実状は依然としてFORTRANを使う人が圧倒的に多く、PL/Iを使う人は極めて少ない。その比は、喫茶店へ入ってコーヒーを注文する人とトマトジュースを注文する人との比よりもはるかに大きい。やや古いデータで恐縮だが、例として表1に1976年2月の東京大学大型計算機センターにおける言語使用比率を示す。

表1. 東大大型計算機センターにおける言語使用比率

(1976年2月ジョブ71,873件)

言 語	使用比率
FORTRAN	97.5%
PL/I	1.1%
アセンブラ	1.1%
COBOL	0.2%
BASIC	0.1%

石田晴久：『FORTRANをめぐるソフトウェア工学的ツール』，

情報処理，Vol. 17, No. 9 (1976) より抜萃。

ところで、京都大学ではPL/Iの歴史も阪大よりもやや古く、センターやメーカー側もPL/Iの普及に力を注いでおり、毎年2回程、3日間のPL/I講習会が開かれているためか、PL/I

ユーザの数はFORTRANユーザの数にははるかに及ばないまでも、無視できない数になってきているようである。筆者も以前その講習会を受けるため、暑い夏の京都に3日間朝早くから通った経験がある。しかし残念乍ら、阪大ではセンターもメーカーもPL/Iの普及に京大程は力を入れていない、と言ったらお叱りを受けるだろうか。細かい事だが、センターが用意するパンチ済みのJCLカードのうちの言語選択カードを見ても京大ではFORTRANとPL/Iの両方が用意されているが、阪大ではFORTRANしか用意されていない。

FORTRANに馴れ親しんだユーザにとってPL/Iは何となくとりつきにくい感じも確かにある。それはPL/IがCOBOL的な面やALGOL的な面を含んでいるからであろう。しかし、PL/Iを一旦マスターするとその、FORTRANとは比較にならない強さ、自由さに驚くであろう。けれども、「プログラムはFORTRANで十分間に合うのに、何で今更別のややこしい言語を新しい本まで買って勉強せんといかんのや、そんなヒマもカネもないわい！」とおっしゃる喰わず嫌いのあなた、あなたのためにこの駄文が書かれていますので、と言えば少々生意気に聞えるが、拙稿の目的はPL/Iを全く知らないFORTRANユーザに、FORTRANと逐一比較し乍ら進む形式でPL/Iとはどのようなものかを理解して頂く事にある。従ってFORTRANをPL/Iにコンバートするような解説となる。それ故、通常のPL/Iの入門書とは一味違ったものとなる。しかし、こういう風に見得を切ったものの、筆者とてさほどPL/Iの使用経験が深いわけではないので、随所で間違いを書くかもしれないが、その時は「アホな奴！」と笑ってお許し頂きたい。

第3章 PL/IのPR

PL/IはALGOL, COBOL, FORTRANの3つの言語が三位一体となって生まれた言語である。従ってこれら3つの言語の持つ長所は殆んど全て包含しているし、特にFORTRAN(JIS7000)の持つ機能は100%包含している。しかも、これら3つの言語が持たない機能も多く備えている。こう書くと良い事づくめのようにだがPL/Iにも弁慶の泣き所はある。それはプログラム言語に限らず、あらゆる工業製品について言える事だが(例えばテレビ等の家電製品を考えるとよい)、便利さと高性能化のため、種々の機能をどんどん付加してゆくと、必然的に複雑化、大型化してゆくことになる。PL/Iのコンパイラが必要とするメモリサイズは、他の3つの言語がいずれも30余K語なのに対して90K語以上と、実に3倍以上ものメモリを必要とする。従ってPL/Iは小型機や中小型機には適用しにくく、無理に適用しようとするれば多くの機能を省略したサブセットとなってしまふ。PL/Iは大型ないし超大型機でこそ本領を発揮できるのである。この意味からも、大型計算機センターでPL/Iが殆んど使われていないという現状は実に宝の持ち腐れと言えよう。尚、最近ではAPLのような、より強力な新しい言語もリリースされてい

るが、どの計算機センターでも使えるという一般性から考えても PL/I の価値は大きいと言える。

断片的になってしまいが、FORTRAN ではできないが、PL/I ではできる事柄のいくつかの例を次に列挙してみよう。

1. 配列の添字の値が 0 や負でもよい。例えば、 $A(-5:5)$ と宣言すると (: コロンは「から」の意味、すなわち、-5 から 5 まで、という意味である)、添字は -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 のいずれかをとりうる。又、配列の添字は、例えば $F=3$, 0 のとき、 $A(F)$ のごとく実数であってもよい。一般的に言って、PL/I では整数型と実数型の区別はないわけではないが、FORTRAN ほど神経質な区別を必要としない。
2. 配列の要素同志の演算が 1 つの文で書ける。(例 1)

$$A = \begin{pmatrix} 1, & 2 \\ 3, & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 8, & 7 \\ 6, & 5 \end{pmatrix} \quad \text{のとき,}$$

$$C = A + B ; \text{ とすると,} \quad C = \begin{pmatrix} 9, & 9 \\ 9, & 9 \end{pmatrix},$$

$$D = A - B ; \text{ とすると,} \quad D = \begin{pmatrix} -7, & -5 \\ -3, & -1 \end{pmatrix},$$

$$E = A * B ; \text{ とすると,} \quad E = \begin{pmatrix} 8, & 14 \\ 18, & 20 \end{pmatrix},$$

$$F = A / B ; \text{ とすると,} \quad F = \begin{pmatrix} 1/8, & 2/7 \\ 3/6, & 4/5 \end{pmatrix},$$

$$G = 3 * A ; \text{ とすると,} \quad G = \begin{pmatrix} 3, & 6 \\ 9, & 12 \end{pmatrix},$$

$$A = 0 ; \text{ とすると,} \quad A = \begin{pmatrix} 0, & 0 \\ 0, & 0 \end{pmatrix}$$

となる。すなわち、配列の要素毎の演算がスカラと同様にできる。但し、上例の E と F の乗算、除算については数学の行列の場合とは異なるから要注意である。あくまで単なる要素同志の演算である。

3. DO 文の各パラメータは実数でも整数でも、両者が混在していてもよい。

(例 2)

```
DO A = -7.5 TO 3 BY 0.5 ;
```

この例は、 $A=-7.5$ から 3 まで、0.5 きざみで DO のループを実行する事である。上記の型は FORTRAN の DO 文に対応するものであるが、これ以外に次のような DO 文もある。

(例3) `DO WHILE (A < 5.5) ;`

`DO WHILE (A < 5.5) ;`

この例ではAが5.5以上にならない限りDOを繰返す。従って、DOのループの中でAが増加していく事が前提となる。

(例4)

`DO A = 1.5, 2.3, 3.1 ;`

この例ではA=1.5, A=2.3, A=3.1 と、3つの場合について実行する。

以上の3種類のDO文は次のようにそれぞれ組み合わせても使用できる。

(例5)

`DO A = -1.5 TO 3.5 BY 0.5, 3.9, 4.7, 6.0, WHILE
(B < 15.5) ;`

この例では、A=-1.5から3.5まで0.5きざみで繰返し、それが済んだら、A=3.9, A=4.7, A=6.0で実行する。但し、 $B \geq 15.5$ となったらDOの実行を打ち切る。

又、DOのパラメータに配列要素名が許される。

(例6)

配列K(2)があるとき、

`DO L=K(1) TO K(2) ;`

と書いてもよい。FORTRANでこれをするとうエラーとなり、どうしてもその必要のある時は、例えばK(1)を変数K1, K(2)を変数K2に、EQUIVALENCE文又は代入文で結びつけておいてから、

`DO 1 L=K1, K2`

としなければならない。

- IF文にTHEN節とELSE節があり、IF文の入れ子ができる。従って、フローチャートで判断のひし型が横にいくつも並んでいるような場合も非常に書き易い。FORTRANのIF文はTHEN節だけでELSE節のない場合に相当する。FORTRANではとかく、GOTO文のかたまりになりがちである。
- ; (セミコロン)で区切る事により、1行にいくつものステップが書ける。これは通常のFORTRANでは許されていないが、ACOS-6 FORTRANだけは多重文として許している。これはALGOLを元祖とするものである。
- 宣言文(DCL文)はプログラムのどこにあってもよい。FORTRANでは宣言文は必ず実行文の前になければならない。
- 再帰的(回帰的)手続きが可能である。すなわち、あるサブルーチン又は関数の中で自分自身

を呼び出す文があってもよい。例えば $N!$ (N の階乗)を次のような関数 $F(N)$ で定義できる。

$$F(N) = N * F(N-1) ;$$

$$(但し、F(1) = 1)$$

再帰的手続きはALGOLのJIS 6000以上で許されているがFORTRANでは不可能である。

8. ON文により、異常事態(割込み)が起った時の処理をプログラマが指定できる。

JIS7000を越えたFORTRANにおける

```
READ( ○ , ○ , END = ○ , ERR = ○ ) ○
```

のENDパラメータとERRパラメータがわずかにこれに相当するが、PL/IのON文ではもっと多種多様な条件文が用意されている。

9. 一般的に言って、PL/IはFORTRANよりはるかに豊富な入出力機能を有している。

以上掲げた9つの項目はPL/IがFORTRANより強力である点のほんの一部分にすぎない。よく言われる事だが、PL/Iは初心者にはやさしく、ベテランには奥の深い性質を持った言語である。FORTRANもその簡潔さという点で初心者にはやさしい言語であるが、PL/Iほど奥深さは感じられない。COBOLは初心者にとってあまりやさしい言語とは思えないが、コンピュータの動作を全体的に把握するのに適している言語であると筆者は思う。

上述の9つの項目を読んで、「ホー、PL/Iというのはそんな事ができるのか、中々便利なものじゃのう。」と、少しでもPL/Iに興味を持たれた方は拙稿を最後まで通読される事を期待する。拙稿は大体4回連載の予定である。

第4章 ACOSのPL/Iについて

ACOSには2種類のPL/Iコンパイラが用意されている。1つは元祖IBM仕様に基づく、ACOS-6 PL/Iであり、もう1つはECMA/ANSIの標準PL/I(BASIS/1)仕様に基づく、ACOS-6 標準PL/Iである。前者はいわば、純アメリカンタイプであり、後者はヨーロッパアメリカンタイプである。ACOSのPL/Iコンパイラとしては後者の方が先輩である。この2つのコンパイラの選択は $\$$ PL1なるJCLのコンパイラオプションのTYPEという項目で選択する。すなわち、ACOS-6 標準PL/Iを使う時は、

```
1 カラム      8 カラム      16 カラム
 $\$$               PL1          TYPE(A)
```

とし、ACOS-6 PL/Iを使う時は、

```
1 カラム      8 カラム      16 カラム
 $\$$               PL1          TYPE(AB)
```


とする。TYPE指定を省略した時のモードは、以前はTYPE(A)になっていたが最近TYPE(AB)に変更されたようである。以下、本稿では両コンパイラをTYPE(A), TYPE(AB)として区別する。余談だが、一方をTYPE(A)としたのなら、もう一方をなぜTYPE(B)としなかったのか不思議である。

両者を比較すると、TYPE(AB)の方が、やや機能が大きいようである。しかし、TYPE(A)の方が古いためか、コンパイラとしては安定しているように筆者には思える。TYPE(AB)の方はまだ手を加えねばならない点が多いように思える。例えば、ラインプリンタの1行の印刷可能文字数はハードウェア的には132文字なのに、TYPE(AB)では標準設定が120文字になっており、121文字以上打たそうとすると勝手に改行されて見苦しいものになってしまう。これは極めて不自然な設定と言わざるを得ない。TYPE(A)では132文字フルに打つ事ができる。このような事から、初心者はTYPE(A)の方を使われる事をお勧めする。PL/Iを使う時のJCLに関する詳しい話はこちらの章とする予定である。

第5章 PL/Iのコーディング

筆者は以前PL/Iの勉強をした頃、PL/I用のコーディング用紙を買おうと思って阪大、京大の生協はじめ、大阪市内の書店、文具店等を捜し回ったけれども、FORTRAN用のコーディング用紙はどこでも置いてあったがPL/I用のコーディング用紙にはついにお目にかかれなかった。これはIBMあたりの会社へ行けばあるのかも知れないが一般の帳簿類メーカーは作っていないのだろう。しかし、よくよく考えてみればPL/IはFORTRANと違って自由形式なので、特別なコーディング用紙は必要としないのである。普通のノート類でもよいし、FORTRANのコーディング用紙を流用してもよい。すなわち、PL/Iでは2カラムから72カラムの間であればどこに何を書いてもよいのであり、FORTRANのように2カラムから5カラムに文番号、6カラムは継続行、7カラム以降に文を書く、といったような制約がないのである。1カラムには何も書いてはいけない。73~80カラムはFORTRANと同じIDENTIFICATION欄なのでプログラムは書けない。FORTRANでは1カラムにCと書く事により注釈行を表わすが、PL/Iでは注釈行という概念はなく、どこにでも注釈をチョイチョイと入れる事ができる。注釈は/*と、*/ではさめばよい。/*KYOU WA YOI OTENKI DA*/という風にすればよい。算術代入文の中に入れてもよい。例えば、

```
K/*KINGAKU*/=T/*TANKA*/*S/*SUURYOU*/;
```

というような書き方ができる。少しややこしくなったが説明的に書くと、

```
K(金額)=T(単価)*S(数量);
```

という事なのである。ここで言い遅れたが、最後の;(セミコロン)はPL/Iでは1つの文のシ

ッポに面倒ではあるが必ず付けなければならないものである。FORTRANに馴れた者がPL/Iのコーディングやパンチを行う際、最も忘れやすいのがこのセミコロンである。注釈に関して1つ注意すべきは、FORTRANの場合、例えば、

```
1 カラム                      72 カラム
C.....A B C D E.....V W X Y Z
```

というように、注釈だと72カラムをはみ出しても別にどうという事もなく、ACOSの場合だとリスト出力の際に72と73カラムの間が少しあいて印刷されるだけである。ところがPL/Iで、これと同様に考えて、

```
2 カラム                      72 カラム
/*.....A B C D E.....V W X Y Z */
```

とするとたちまちエラーになる。すなわち、73カラム以降は何も書いていないのと同じなので、注釈の始めを示す/*があるのに、終りを示す*/がなく、次の行、その次の行と、どんどん*/を捜し求める事になり、プログラムは支離滅裂となる。

コーディングでもう一つ注意すべき事は空白の使い方である。FORTRANでは空白はどこに入れてもよかったが、PL/Iではそれ程自由ではない。例えばFORTRANで、RETURNと書く所を気取って、

```
R      E      T      U      R      N
```

などと書く人がいるが、PL/Iでこれをするるとたちまちエラーになる。しかしPL/Iでも、

```
A=B+C ; を、
```

```
A = B + C ;
```

とする事は差支えない。すなわち、キーワードや変数名等、プログラムの一つの要素となっているものを更に空白で切る事は許されないが、=や+の両側のように論理的に切れている所には空白をいくつ入れてもよい。ACOSではプログラムの最初を示す、PROC ;を、

```
P R O C ;
```

とすればコンパイルもされずにエラーとなる。

今回はPL/Iのほんの入口を御紹介しただけに止ったが、次回は宣言文や入出力文、代入文等について述べる予定である。では又次号でお目にかかりましょう。サヨナラサヨナラサヨナラ

(プログラム相談員)