高級言語パスカルによるソフトウェア開発

名古屋事業所　開発部　岡　田　勝　男
〃　　〃　広　瀬　英　雄

内　容　梗　概

オレゴンミニコンピュータ社（ＯＭＳＩ）の開発によるパスカルコンパイラを導入し，当社設置のプログラム開発用ミニコンＰＤＰ11/34（ＯＳはＲＳＸ—11/M）のもとで，言語パスカルによるプログラムの開発を開始し，ソフトウェア開発に実用的に採用してゆくメリットが得られたので報告する。

1. まえがき

一般に，プロセスコントロールシステムにおいて，ソフトウェアのプログラミングにかかる比率は大きい。例えば，あるプロセスコントロールシステムにおいては，それは工数，期間についてそれぞれ1/2，1/3の割合を占める。したがって，適切な高級プログラミング言語を選定することによって次のようなメリットが得られる。

1) アセンブリ言語のように，マシンデペンデントでなくソフトウェア互換性が可能となり，ソフトウェア技術の蓄積化が可能となる。
2) プログラム作成期の能率向上により，ソフトウェアチームの少人数化ができ，プログラマ間のオーバヘッドが減少し，相互のインタフェイスが少なくなり，トータルの工数削減のみでなくソフトウェアの品質が向上する。
3) プログラム手法の標準化が可能となり，ソフトウェアの品質が安定する。

しかし，一般に高級言語で実行させるとスピードが落ちる傾向があった。これに対して，最近プロセスコントロールにおいても信頼性が高く，実行効率の良い高級言語によるプログラミングの開発が提唱されてきている。当社においてもこのような状況をかんがみ，高級言語の一つであるパスカルによってプログラムの開発を開始することにした。

その一環として今回は，オレゴンミニコンピュータ社の開発したシーケンシャルなパスカルのコンパイラを使って技術計算プログラム開発への応用を試みた結果，データ構造の表現，系統的プログラミングの開発に優れた威力を発揮することが分かった。

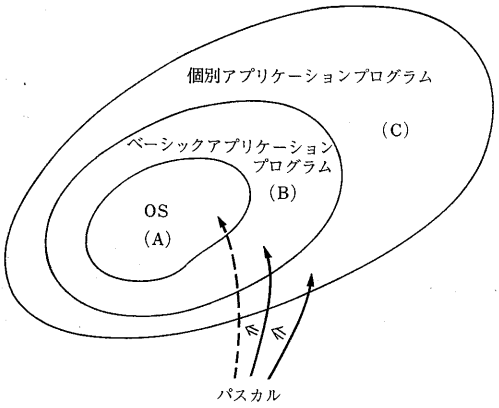
2. システムと言語

パスカルは，フォートラン，コボルと同様の高級言語の一種ではあるが，単にアプリケーションプログラムを作成する言語にとどまっていない。それは，ソフトウェアのアルゴリズムを記述する言語として採用され，広く応用範囲を持つものである。例えば1975年，南カリフォルニア大学のブリンク・ハンセンは，DEC 社の PDP 11/45 のマシンを使ってコンカレント・パスカルプログラムの実行をソロオペレーティングシステムの形で実現している。しかも，マシンに従属しているアセンブリ言語はわずか４％しか使われない。このことはこのオペレーティングシステムが他のマシンに容易に移植可能であることを示すものである。

また，同じくハンセンの設計したスケジューラーのプログラムはアセンブリ言語だと１年，パスカルだと１週間で開発したという報告がでているが，このプログラム開発工数の減り方には目を見張るものがある。この他，パスカルは同様なシステム記述用の言語である，ユークリッド，モジュラなどの言語にも影響を与えている。

現在，プロセスコントロールシステムを記述している言語は，アセンブリ言語であり，そのソフトウェア構成は一般的に第１図のようになっている。

ここに開発工数としては，（Ａ），（Ｂ），（Ｃ）の部分いずれも大差はないが，単位時間あたりのプログラムの使用頻度は圧倒的に（Ｃ）の部分が少なく，しかもこの部分は高速化を要求されることはないので，一般にアセンブリ言語で記述されることはメモリーの節約化以外に



第1図　ソフトウェア構成図

メリットがない。したがって，アプリケーション部分はすぐにでも高級言語化すべき部分であって，これを果たすことによって，現在行なわれている詳細設計の工数も極端に少なくなり（あるいは全くなくなることもありうる），詳細設計書の形式も全く変わって，アルゴリズムが前面に浮き出してくると考えられる。

また，上述したようにアプリケーション部分の高級言語化が終わると，ベーシックアプリケーション部分，OS部分へと，その影響は波及することになり，ソフトウェアが明確に記述されることになる。

3.　パスカルの概要

3.1　標準パカスルについて

スイス ETH のニコラウス・ヴィルト設計によるプログラミング言語パスカルは，1974年「PASCAL USER MANUAL AND REPORT」の形で標準パスカルとして紹介され，広く世に普及するようになった。

元来パスカルは，ヴィルトがアルゴル60の後継言語として IFIP（情報処理国際連盟）に提案した言語であり，アルゴル60の流れをくみ数値計算のみならず，情報検索や記号処理の分野にまで処理を広げようと設計されたものであった。この狙いのとおり，パスカルは極めて堅固なデータ構造を持ち，系統的プログラム開発が可能な言語構造がとられ，文法は明快である。したがって信頼性があり，実行効率の良い処理系を作ることが可能であるという特徴をもっている。

3.2　標準パスカルの文法

（1）　プログラムの構造

プログラムは，ブロックから構成されブロックは宣言部と実行文部から成る。宣言部は，ラベル宣言部，定数宣言部，型定義部，変数宣言部，手続き関数宣言部に分かれる。

実行文部は，begin ＜文＞；＜文＞；…＜文＞end.の形をとる。したがって，プログラムは入れ子構造を許しトップダウン方式のプログラム開発が容易になる。

（2）　制御構造

制御構造は，大きく分けて条件文と繰り返し文から構成される。

条件文は if 文（if ＜論理式＞then＜文＞elese＜文＞）と，case 文より成り，繰り返し文は制御変数を持った for 文，論理判断による repeat 文, while 文から成る。このような制御構造は，構造的プログラミングを可能としている。

（3）　データの型および構造

基本的なデータ型は，整数，実数，論理値，文字の4種類であるが，これらを組み合わせてユーザが勝手にデータの型を定義することができる。

(a)　単純型

上記の4つの基本型に加え，とりうる値全体を羅列することによって定義される型をスカラー型とよび，その部分範囲とあわせて単純型という。

(b)　構造型

構造型は，要素を持つようなデータの型で，要素からの構成方法により配列型，レコード型，集合型，ファイル型の4種類に分かれる。

配列型はフォートランのディメンション文に相当するものであり，レコード型は異なる型のデータを幾つか，ひとまとめにしたものであり，集合型は集合の演算が可能な単純型のデータの集まりであり，ファイル型は，いわゆる順編成ファイルの概念を論理的に整理したものである。

このように，データの型及び構造は，他言語に見られないほど豊かな内容を持っているので，データのドキュメント化には特に有効である。また，プログラマは最初から，堅固なデータ構造の設計を余儀なくされるので必然的に曖昧さを避けられ，信頼性の高いプログラムを作ることができる。

（4）　手続き（関数）

パスカルの特徴的な手続きの性質は，再帰呼び出し（自分自身で自分を呼び出す）ができることである。このことにより，プログラムは見やすくなり，したがって，プログラムの信頼性が高くなる。また，再帰的な性質は手続きのみならず，データの取り扱いにも適用され，データはダイナミックな構造をとることを可能にしている。更に，手続きの引数は定数と変数の2種類をもち，手続き用のデータの受け渡し（インタフェイス）をはっきりと定義しているため，インタフェイスミスが少なくなる利点がある。

3.3　パスカルコンパイラ

パスカルコンパイラが出したエラーメッセージを第2図に示す。メッセージによるエラーの内容は他言語に見られないほど明快となっている。コンパイル段階でエラーメッセージが除去されると，作り上げられたプログラ

```
.MAIN.                OMSI PASCAL-1  RSX  V1.1G 28-Nov-79 15:05  Page 3
TAKAOKA SEISAKUSYO NAGOYA JAPAN

LINE  STMT LEVEL  NEST   SOURCE STATEMENT

 109
 110
 111
 112
 113
 114                      /*$E+
 115                       TCR:TRANSPOSE COMPLEX TO REAL
 116                       */
 117
 118                    VAR
 119                      r:integer;
 120                      PROCEDURE tcr(a:complex; VAR m:real; VAR p:real);
 121
 122                       BEGIN
 123     1     2     1       m:=sqrt(sqr(a[1])+sqr(a[2]));
 124
 125     2     2     1       IF ((a[1]>0) AND (a[2]>=0))
 126     3     2     2        THEN p:=(180/pi)*arctan(a[2]/a[1]);
 127     4     2     1       IF ((a[1]<0) AND (a[2]>=0))
 128     5     2     2        THEN p:=(180/pi)*arctan(a[2]/a[1])+180;
 129     6     2     1       IF ((a[1]<0) AND (a[2]<=0))
 130     7     2     2        THEN p:=(180/pi)*arctan(a[2]/a[1])+180;
 131     8     2     1       IF ((a[1]>0) AND (a[2]<=0))
 132     9     2     2        THEN p:=(180/pi)*arctan(a[2]/a[1])+360;
 133    10     2     1       IF ((a[1]= 0) AND (a[2]>0))
 134    11     2     2        THEN p:=90;
 135    12     2     1       IF ((a[1]= 0) AND (a[2]<0))
 136    13     2     2        THEN p:=270;
 137    14     2     1       IF ((a[1] =0) AND (a[2]=0))
 138    15     2     2        THEN p:=0;
 139
 140    16     2     1       r:=p*(pi/180);
                                          ^
******          INCOMPATIBLE TYPE.

 141    17     2     1     END;
******  FATAL ERROR:    MISSING END
ERRORS DETECTED:  2
FREE MEMORY:  3246  WORDS
```

エラー箇所

エラーメッセージ

第２図　コンパイルリスト

ムのタスクイメージは即座に実行可能となり，ほとんど不注意によるプログラムミスは除かれる。したがって，プログラマはコンパイル段階で苦しむだけで良く，デバッグの期間が短かくなる。

4. OMSI パスカルについて

標準パスカルを PDP—11 で実行できる環境を作るため，OMSI パスカルは標準パスカルに加え，次のような機能を備えている。

（１） マクロコードの挿入（PDP—11用）。

（２） 外部手続き，フォートランの呼び出し機能。

（３） グローバルシンボルの参照機能。

（４） PDP—11用の外部ファイルアクセス手続きの追加。

したがって，プログラムはモジュールごとの開発が可能であり，すでに開発されたフォートランサブルチンパッケージを加えることも可能である。

5. パスカルの効率

パスカルの効率を調べるため簡単なテストを行い，フォートランと比較してみた。その結果を第１表に示す。ここで実行効率は第１表に示す８個の組み込み関数を，

第１表　パスカルの効率

(a) メモリー効率

組み込み関数	パスカル(ワード)	フォートラン(ワード)
絶対値（実数型）	7808	4416
絶対値（整数型）	7808	4384
平方根	7840	4416
指数	7904	4480
自然対数	7872	4448
正弦	7904	4448
余弦	7904	4448
逆正接	7872	4512

ただし

	パスカル	フォートラン
HEAP	2000	——
FSR	1080	1080
RESL	2926	563

(b) 実行効率

	パスカル（秒）	フォートラン（秒）
実行時間	24.6	38.3

```
.MAIN.              UMSI PASCAL-1  RSX  V1.10  9-Nov-79 11:23  Page 3
TAKAOKA SEISAKUSYO NAGOYA JAPAN

LINE  STMT LEVEL  NEST   SOURCE STATEMENT

 109
 110
 111
 112
 113
 114                        /*
 115                         MAIN PROGRAM OF CABIN
 116                         */
 117                        PROCEDURE readdata;external;
 118                          PROCEDURE specification(n:integer);external;
 119                            PROCEDURE initialize;external;
 120                              PROCEDURE proceed;external;
 121                               BEGIN
 122     1     1     1                reset(inf,inputfile);
 123     2     1     1                rewrite(out,outputfile);
 124     3     1     1                readdata;
 125     4     1     1                IF spec<10
 126     5     1     2                 THEN specification(spec)
 127                                    ELSE
 128     6     1     2                  BEGIN
 129     7     1     3                    initialize;
 130     8     1     3                    proceed;
 131     9     1     3                  END;
 132    10     1     1              END.

 ERRORS DETECTED:  0
 FREE MEMORY:  3166  WORDS
```

第3図　プログラム CABIN

0^4 回繰り返し実行させたときの実行の時間を測定したものである。このテストに限り，パスカルはフォートランよりも実行速度が速く，メモリー効率は同程度だということができる。

ここに HEAP は，スタックマシンを意識した言語パスカルに特有な概念であり，一種のバッファを表わす。また，FSR，RESL は RSX-11/M に特有な一種のファイルバッファである。

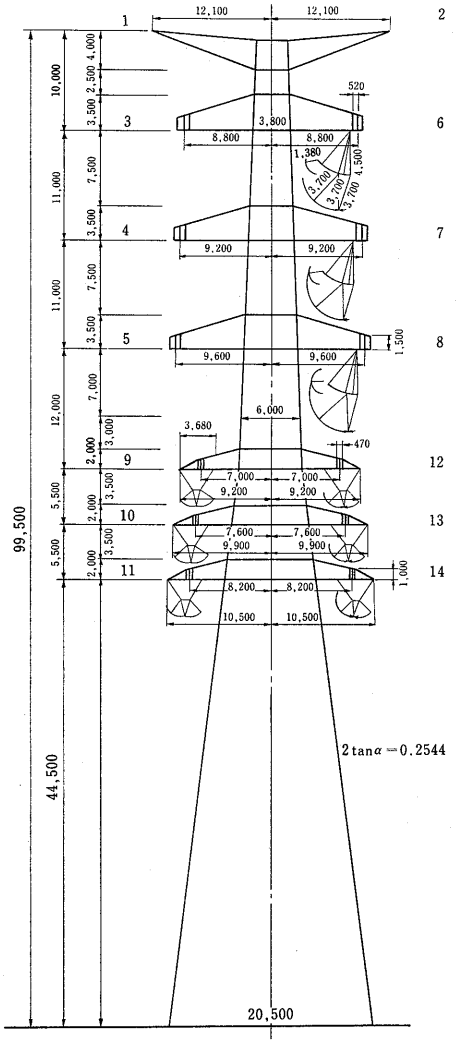
6. プログラム CABIN の開発

プログラム CABIN は，高圧架空送電線の活線による死線の誘導電圧電流を計算するものである。

第2表　計　算　条　件

線番号	電圧 (kV)	電流 (kA)	位相 (度)	両端状態
1				接　地
2				接　地
3				開　放
4				開　放
5				開　放
6	$500/\sqrt{3}$	13.78	120	開　放
7	$500/\sqrt{3}$	13.78	0	開　放
8	$500/\sqrt{3}$	13.78	240	開　放
9	$154/\sqrt{3}$	4.6	0	開　放
10	$154/\sqrt{3}$	4.6	120	開　放
11	$154/\sqrt{3}$	4.6	240	開　放
12				接　地
13				接　地
14				接　地

併架亘長 6.9 (km)



第4図　500kV/154kV 併架送電線鉄塔モデル図

CABIN : POWER CABLE INDUCTION PROBLEM (VERSION 1 , AT 10/79) DATE: 79/11/22 PAGE: 3

BY FEIC SYSTEM ** TAKAOKA MFG. CO., LTD. NAGOYA ** -SOFTWARE LABORATORY-

* PATTERN = EXAMPLE-(500KV/154KV:PARALLEL)

***** ｹｲｻﾝ ｹｯｶ *****

LINE	SITUATION	VOLTAGE(MAGNITUDE,PHASE) (REAL ,IMAGINAL)		CURRENT(MAGNITUDE,PHASE) (REAL ,IMAGINAL)		INDUCTION
1	DEAD	2.955908E+03	353.8	2.513334E+02	264.1	MAGNETIC
		2.938457E+03	-3.207205E+02	-2.582028E+01	-2.500036E+02	
2	DEAD	5.905853E+03	5.1	4.308123E+02	275.1	MAGNETIC
		5.882804E+03	5.212564E+02	3.829730E+01	-4.291066E+02	
3	DEAD	3.021912E+04	87.1	0.000000E+00	0.0	STATIC
		1.532192E+03	3.018025E+04	0.000000E+00	0.000000E+00	
4	DEAD	1.550448E+04	32.2	0.000000E+00	0.0	STATIC
		1.311918E+04	8.262927E+03	0.000000E+00	0.000000E+00	
5	DEAD	1.525588E+04	318.1	0.000000E+00	0.0	STATIC
		1.135919E+04	-1.018384E+04	0.000000E+00	0.000000E+00	
6	LIVE	2.887000E+05	120.0	1.378000E+04	120.0	
7	LIVE	2.887000E+05	0.0	1.378000E+04	0.0	
8	LIVE	2.887000E+05	240.0	1.378000E+04	240.0	
9	LIVE	8.889999E+04	0.0	4.600000E+03	0.0	
10	LIVE	8.889999E+04	120.0	4.600000E+03	120.0	
11	LIVE	8.889999E+04	240.0	4.600000E+03	240.0	
12	DEAD	4.974001E+03	176.5	1.074502E+03	95.0	MAGNETIC
		-4.964531E+03	3.067856E+02	-9.351597E+01	1.070425E+03	
13	DEAD	4.026018E+03	174.2	4.687805E+02	90.8	MAGNETIC
		-4.005256E+03	4.083384E+02	-6.526009E+00	4.687351E+02	
14	DEAD	3.527733E+03	170.9	3.537031E+02	78.7	MAGNETIC
		-3.483485E+03	5.569849E+02	6.903764E+01	3.469001E+02	

第5図 計 算 結 果

入力データは，送電線形態の幾何学的形状，送電線の抵抗率，半径等の基礎定数，送電線電圧，電流である。出力は，各線路のインピーダンス，誘導電圧，電流である。

計算途中必要な回路方程式は，入力データを基に計算機が自動的に作成するもので，キルヒホフの閉路方程式，節点方程式による。また，線路の自己インダクタンス，相互インダクタンスの式はカーソン・ポーラチェックの式による。

プログラムはほとんどパスカルで書かれており，わずかに線型方程式を解く部分のみがフォートランで書かれている。総ステップ数は約1000ステップ，メモリーサイズは32kW，開発工数は設計も含めて1人月である。プログラム CABIN の一部を第3図に示す。

次に，実際の計算例を示す。ここに第4図は500 kV/154kV併架送電線鉄塔のモデル図，第2表は計算条件，第5図は計算結果のリストである。

7. あ と が き

アセンブリ言語によるプログラミングというのは，それが手作業であるという性格上，生産性が悪い。そこで，これを改善するため，言語を高水準にしてコンピュータシステムを作るという発想は，すでに定着しつつあり，パスカルはその起爆剤になったという気がする。

今回，シーケンシャルプログラムを開発したことにより，パスカルの特徴をとらえることができ，実用化のめどがついた。確かに，シーケンシャルパスカルだけではプロセスコントロールを記述することは不十分であるが，並行処理のプログラムでもシーケンシャルパスカルの占める割合は大きく，パスカルの思想を得るということは重要な意味がある。そこで今後は，アセンブリ言語の器用さに強力に迫るプロセスコントロール用高級言語，コンカレントパスカル，モジュラなどの実用化に向けて研究を進めたい。

終わりに，今回の開発にあたり，ご指導ご協力いただいた名古屋大学情報工学科の吉田助教授をはじめ関係各位に深謝する。

参 考 文 献

(1) Jansen, Wirth: Pascal User Manual and Report, Springer Study Eddition

(2) Hansen: The Architecture of Concurrent Programs, Prentice-Hall

(3) Wirth: The Use of Modula, Software-Practice and Experience, Vol. 7, (1977), pp. 3-35.

(4) 和田英一：プログラム言語 Pascal bit Vol. 10, No. 1～No. 10