PASCALプログラムの相違点

藤木健士*

新システムのＳｕｎワークステーション上では Sun PASCAL が使用可能であり，プログラミング演習等でこれを用いることができる．ここでは，従来から多く用いられている Turbo PASCAL と Sun PASCAL との主な相違点について述べる．

１．使用環境

新システムで提供される環境は Turbo PASCAL のような統合環境ではなく，ＵＮＩＸ上でコマンドを実行することにより，ソースプログラムの作成，コンパイルおよび実行の各々の作業を行うことになる．プログラムの作成は emacs（「基本的な使用法(1)」を参照）等のエディタを用いて行う．この時，PASCAL のソースプログラムのファイル名は *.p あるいは *.pas でなければならない点に注意しなければならない．ＩＢＭ４３８１やＩＢＭ３０８１からプログラムを移行する場合，ファイル名がこの規則に従うように変更する必要がある．ソースプログラムが完成したらそれをコンパイルすることになるが，コンパイルは次のコマンドを実行することにより行う．ここでは sample.p というソースファイルをコンパイルする例を示す．

```
pc  sample.p
```

ソースプログラムがコンパイルされ，エラーなく終了すると a.out という実行ファイルが生成される．実行ファイルを a.out 以外の名前で作成したい場合には，-o オプションを用いて次のように入力する．

```
pc  -o  sample  sample.p
```

* 情報科学センター　fujiki@isci.kyutech.ac.jp

作成された実行ファイルのプログラムを実行するには，次のようにその実行ファイル名をコマンドとして入力する．

```
a.out
```

実行ファイルプログラムのデバッグにはシンボリックデバッガ dbx を用いることができる．ただしコンパイル時に -g オプションをつけて生成した実行ファイルに限る（基本的な使用法(I)を参照）．dbx の詳しい使用法については Sun のマニュアル 'Debugging Tools' を参照されたい．

２．言語仕様の相違点

Turbo-PASCAL は PASCAL の標準のＩＳＯ規格に準拠して作られているわけではない．それに対して，Sun Pascal は他の言語とのリンクや処理系依存の部分を除いて，標準 PASCALに比較的近い処理系になっている．ここでは，これら２つの言語処理系の間の言語仕様の主な相違点について述べる．

(1) プログラム文を必ず記述する必要がある．

Sun Pascal ではプログラム先頭のプログラム文を省略できない．

(2) case 文のケースラベルは定数でなければならない

Turbo-Pascal では case 文のケースラベルは変数でもよいが，Sun Pascal では定数でなければならない．例えば Turbo-Pascal では次のように記述できる．

```
case i of
   j: result := 3*i;
   k: result := 2*i;
   l: result := 5*i;
end;
```

しかし，Sun Pascal では下のようにケースラベルを定数で記述する．

```
case i of
   1: result := 3*i;
   2: result := 2*i;
   3: result := 5*i;
end;
```

どうしてもケースラベルに変数を用いなければ処理できない場合には，if 文で以下のように記述すれば同じ意味の文を書ける．

```
if i = j then
   result := 3*i
else if i = k then
   result := 2*i
else if i = l then
   result := 5*i;
```

(3) 識別子は大文字小文字を区別する

Sun Pascal では変数名や関数名などの識別子で用いる英字の大文字と小文字（例えば I と i）とを同一視しない．また，予約語や事前に宣言された識別子はすべて小文字を使わなければならない．例えば，次のプログラムをコンパイルしたとする．

```
program samp(input,output);
BEGIN
        writeln('hello')
end.
```

このプログラムのコンパイルは失敗する．これは BEGIN が大文字であるために予約語の begin とみなさずにエラーとなってしまうからである．このため予約語はすべて小文字を

使用する必要がある．また事前に宣言された識別子（宣言しなくても使用できる定数，型，変数，手続きおよび関数の識別子）もすべて小文字で宣言または定義されているのでプログラム中で使用するときも小文字を使わなければならない．

またユーザが宣言または定義した識別子も大文字と小文字の区別を行うので注意しなければならない．次に例を示す．

```
program test(output);
var     i:integer;
begin
        I:=10
end.
```

このプログラムをコンパイルすると，’４行目で識別子が宣言されていない’とメッセージが出されコンパイルが失敗する．これは小文字の 'i' と大文字の 'I' を区別するので 'I' が宣言されていないとみなされるためである．それで上のプログラムの４行目を 'i:=10' に書き換えるとコンパイルできる．

(4) ファイル入出力

turbo-Pascal ではファイル変数にファイルを割り当て，その後 reset や rewrite手続きを行ってファイルのオープンを行うと，ファイルの入出力が可能となる．次にその例を示す．

```
        assign(fp,'input.dat');
        reset(fp);
```

ここでは assign 手続きでファイル変数 fp にファイル input.dat を割り当て，reset 手続きでファイルをオープンしている．Sun Pascal では，これを同じ内容を次のように１行で記述する．

```
        reset(fp,'input.dat');
```

rewrite 手続きもファイルのファイル変数への割当とオープンの両方の機能をもち，次のように記述する．

```
rewrite(fp,'result')
```

(5) １６進数が使用できない

Turbo-Pascal では $1E のように１６進定数を使用することができたが，Sun Pascal では，１６進定数が使用できない．プログラムを移行する場合には１０進数に書き直す必要がある．

(6) string 型が使用できない

標準 Pascal では文字列型の変数の定義は次のように記述する．

```
var     name:packed array[1..14] of char;
```

Turbo-Pascal ではこの形式の他に次に示すように string という型を設けてよく使う文字列型を簡単に書けるようになっている．

```
var     name:string[14];
```

Sun Pascal では varying という特別な配列型を用いて，これと似た形で文字列の宣言をすることができる．

```
var     name:varying[14] of char;
```

(7) 使用できない演算子がある

Sun Pascal には shl（右シフト）や shr（左シフト）といったシフト演算が使用できない．また論理演算 xor（排他的論理和）も使用することができない．しかしこれらは同じ

動作をする関数を使用できるので，これらの演算子は以下のように書き換えるとよい．

```
X shr N        ------->      rshft(X,N)
Y shl M        ------->      lshft(Y,N)
A xor B        ------->      xor(A,B)
```

(8) GOTO 文でブロックから飛び出すことができる

turbo-Pascal では GOTO 文でブロックから外に制御を移すことができなかったが，Sun Pascal ではこれが可能である．

(9) 手続き引数、関数引数を用いることができる

turbo-Pascal では使用できない標準 Pascal 仕様の手続き引数および関数引数を Sun Pascal では使用することができる．

(10) ファイルのインクルード

ファイルのインクルードは Turbo-Pascal では次のように記述する．

```
{$I  filename}
```

これにより filename で指定したファイルをこの位置に展開する．Sun Pascal ではこれを次のようにＣ言語と似た形で記述する．

```
#include 'filename'
```

以上で turbo-Pascal と Sun Pascal の主な違いを述べた．ここでは主なものに関してのみ説明したので，詳細を知りたい場合には Sun Pascal Reference Manual を参照されたい．

3．定数，型，変数，関数および手続きの対応表

以下に標準で提供されている定数，型，変数，関数および手続きに関して Turbo-Pascal と Sun Pascal との対応表を示す．ここでは主なもののみを示す．

Turbo Pascal	種類	Sun Pascal
Abs(r)	関数	abs(r)
ArcTan(r)	関数	arctan(r)
Assign(fp,s)	手続き	なし，reset および rewrite がこの機能を持つ
Boolean	型	boolean
Byte	型	なし
Char	型	char
Chr(i)	関数	chr(i)
Close(fp)	手続き	close(fp)
Concat(s,s...)	関数	なし，文字列に関しても + 演算が使用できる．
Copy(s,p,n)	関数	substr(s,p,n)
Cos(r)	関数	cos(r)
Delete	手続き	なし
EOF,EOF(fp)	関数	eof,eof(fp)
EOLN,EOLN(fp)	関数	eoln,eoln(fp)
Exit	手続き	なし
Exp(r)	関数	exp(r)
False	定数	false
Flush(fp)	手続き	flush(fp)
Frac(r)	関数	なし
Input	変数	input
Insert	手続き	なし
Int	関数	なし
Integer	型	integer
Length(s)	関数	length(s)

Ln(r)	関数	ln(r)
MaxInt	定数	maxint
New(x)	手続き	new(x)
Odd(i)	関数	odd(i)
Ord(e)	関数	ord(e)
Output	変数	output
Pi	定数	なし
Pos(s1,s2)	関数	index(s1,s2)
Pred(e)	関数	pred(e)
Random(i)	関数	random(i)
Randomize	手続き	なし
Read(....)	手続き	read(....)
ReadLn(....)	手続き	readln(....)
Real	型	real
Reset(fp)	手続き	reset(fp,<file_name>)
Rewrite(fp)	手続き	rewrite(fp,<file_name>)
Round(r)	関数	round(r)
Sin(r)	関数	sin(r)
Sqr(r)	関数	sqr(r)
Sqrt(r)	関数	sqrt(r)
Str(r,s)	手続き	なし
Succ(e)	関数	succ(e)
Text	型	text
True	定数	true
Trunc(r)	関数	tranc(r)
UpCase(s)	手続き	なし
Val(s,r,f)	手続き	なし
Write(....)	手続き	write(....)
Writeln(....)	手続き	writeln(....)