

TSS 環境における Pascal の幾つかの教具†

宇津宮 孝一^{††} 荒牧 重登^{††}
樽 美和 幸^{†††} 吉田 将^{††††}

近年多くの教育機関において、プログラミングの教育を行うために言語 Pascal が採用され広く使用されている。TSS 環境下で Pascal を情報処理教育等の一環として利用するに当っては、効果的な教具を用意して併用すれば、言語 Pascal が本来もつ優れた教育性と相まって集団教育および個別学習時の教育効果は著しく高められる。本論文では、このような目的で設計・製作し現在教育現場で利用している次の教具について述べる。

(1) 個別学習時には Pascal の構文の学習もでき、集団教育時にはプログラムの実行演示ができる Pascal 構文学習・実行演示システム

(2) 視覚を通じての理解を容易にするため、Pascal に図形作画機能を取り入れた Pascal 図形作画支援システム

(3) 特に個別学習や試験に用いて、理解状況を把握するための Pascal 演習レポート処理支援システム
これらの教具を教師にも学生にも使いやすい形で提供することにより、言語 Pascal を通じて情報処理に対する理解を一層深めることができる。

1. はじめに

開発の動機の一つがプログラミング教育であるとされているプログラミング言語 Pascal¹⁾ は、今日、教育・研究分野で広く使用されるようになってきている。

一方、多くの教育機関で TSS を主体にした情報処理教育等が実施されかなりの成果を得ている。TSS 環境下で Pascal を情報処理教育等の一環として使用することは、教育環境としては言語に備わった教育性と TSS がもつ応答性の良さという点で整っている。しかしながら、これら二つの長所を生かし、教育をさらに充実・深化させ、教育効果を高めるためには、直接的に有効となるような、あるいは触媒としてその効果を発揮するような教具が不可欠である。

本論文ではこのような観点に立って、我々が ACOS-6 オペレーティング・システムに現在までに実現してきた言語 Pascal のための幾つかの教具について述べる。これらの教具は主として次の三つのシステムから成る。

† Some Teaching Tools for Pascal in TSS Environment by KOUICHI UTSUMIYA, SHIGETO ARAMAKI (Educational Center for Information Processing, Kyushu University), KAZUYUKI TARUMI (Department of Computer Science and Communication Engineering, Kyushu University) and SHO YOSHIDA (Department of Electrical Engineering, Kyushu University).

†† 九州大学情報処理教育センター

††† 九州大学工学部情報工学科

†††† 九州大学工学部電子工学科

(1) Pascal 構文学習・実行演示システム²⁾

個別学習時には構文の学習や実行の流れの理解に、集団教育時にはスクリーン・ディスプレイを用いることにより実行演示に有効となるシステムである。

(2) Pascal による図形作画支援システム³⁾

Pascal の良さを保ちながら図形作画機能を実現したもので、視覚を通じての理解を促進し、より広い問題解決への適用を図ることができる。

(3) Pascal プログラム演習レポート処理支援システム⁴⁾

個別学習の成果をレポートとして提出させ、評価する過程を電算化したシステムで、従来手作業で行われていた手間を軽減し、いつでも必要な時に評価が可能なので、学生の理解度を的確に把握することができる。このシステムは通常の試験にも適用できる。

2. Pascal プログラム実行演示システム

2.1 システムの特徴

Pascal プログラム実行演示システム (PASVID: PAScal Visual Demonstration system) を情報処理教育等にも活用することにより次のような教育効果が期待できる。

(1) プログラムが実際に動いている様子を実時間で追うという体験を通じて、プログラミングの生きた学習ができる⁵⁾。

(2) プログラムの実行の流れの実時間表示に加えて、言語の基本的構文を併せて表示するので、文法の

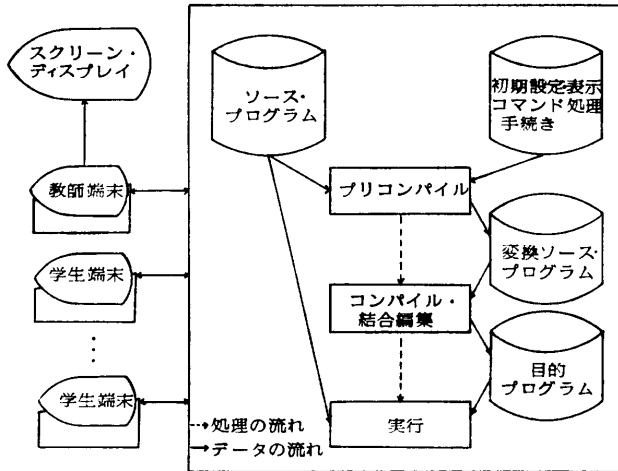


図 1 PASVID システムの概要
Fig. 1 System organization of PASVID.

習得や各実行文の理解が容易になる。

以上は特に個別学習における利点であるが、集団教育においても、次の利点がある。

(3) たとえば、表示装置として大形画面をもつスクリーン・ディスプレイ装置を用いれば、教師が学生の前で演示したり、多人数でプログラム実行中の検査を行うことができる。

このようにして従来暗箱の中に隠れていたプログラムの実行過程を動画を見るような状況で体験できることは、初心者にとってプログラムの神秘性を払しょくし、正しく、効率よく、分かりやすいプログラムの重要性を認識させる点において非常に教育的である。

2.2 システムの構成

システムの使用環境および構成を図 1 に示す。図に示すように PASVID は、牛島らの FORDAP⁶⁾ ですでに実績のあるプリコンパイル方式を採用し、PASVID プリコンパイラにおいて、対象となる Pascal のソース・プログラムに、

- (1) 初期設定の手続き
- (2) 種々の表示のための手続き
- (3) 後述するコマンドの処理手続き
- (4) (1)~(3)の呼出し文

を挿入して、変換されたプログラムを Pascal コンパイラに渡す。実行時には、PASVID は、ソース・プログラムをファイルから読み込み、指定されたコマンドに応じて実行と表示を行う。

2.3 システムの機能

PASVID においては、与えられた Pascal プログラムをどのようなモードで、どのようなコマンドの制御

に基づいて、どのような表示を行いながら実行するかが重要である。これらの機能について説明する。

2.3.1 実行モード

PASVID には、manual モード、automatic モード、および blind モードの三つの実行モードが用意されている。

manual モードは、特定のキーを押すたびに 1 実行制御単位 (ステップと呼び 2.3.3.2 で定義する) ずつ実行し表示を行う。automatic モードは、自動的に 1 ステップずつ実行しながら表示を行い、指定されたステップ数または指定された行番号に達するまで続ける。

これらの二つのモードは次に示す種類の表示を行う。

- (1) 実行しようとする文とその前後数行の文
- (2) その文の基本的構文 (選択機能)
- (3) 実行によって参照された変数の値 (選択機能)

一方、blind モードでは上記の表示を行わずに指定ステップ数または指定行番号に達するまで実行を行う。このモードを使用すれば実行速度を速めることができるので、たとえば manual または automatic モードから切り替えて、ループ外へ速く抜け出しほかの部分の実行をしたい場合等に有効である。

2.3.2 コマンド

2.3.2.1 起動コマンド

ACOS-6 TSS コマンド入力モードで次に述べる指定により PASVID を起動する。以下△は 1 個以上の空白を、_ は入力すべき部分を、Ⓢ は復帰改行を、{ } は列挙した項目の選択を、[] は項目の省略が可能であることをそれぞれ意味するものとする。

$$*PD_{\Delta} \left\{ \begin{array}{l} * \\ \text{ソース・ファイル名} \end{array} \right\} [\Delta \text{オプション}]_{\text{Ⓢ}}$$

ここで、最初の“*”はコマンド入力促進記号を、オペランド中の“*”は一時的テキスト・ファイルを表す。

PASVID が起動されると図 3 の①、②に示した表示を行った後、次節で述べる実行時コマンド待ちとなる。

2.3.2.2 実行時コマンド

実行時コマンドは PASVID からの入力促進記号“?”に続けて、次のようにコマンド名およびオペランドを入力する。

$$? \text{コマンド名} [\Delta \text{オペランド}]_{\text{Ⓢ}}$$

表 1 に実行時コマンドの入力形式と機能・意味につ

表 1 PASVID 実行時コマンド
Table 1 Run-time commands of PASVID.

種別	コマンド名	入力形式	機能・意味
実行制御コマンド	S(Single step)	S	1ステップ実行し、表示を行う。
	A(Automatic)	A Δ {#行番号 ステップ数}	指定の行番号又はステップ数に達するまで実行・表示を行う。
	B(Blind)	B Δ {#行番号 ステップ数}	Aコマンドとの違いは表示をしない点である。
	E(End)	E	実行を終了させる。
表示・変数制御コマンド	D(Display)	D Δ 変数名 注)	指定された変数名の値を表示する。
	R(Refer)	R Δ {+ -}	+を指定するとこれ以降参照が起った変数の値を表示する。 -を指定すると表示しない。
	C(Change)	C Δ 変数名 注)	変数の値を変更する。変更する値は、次行"="の出力に続いて入力する。
	F(Format)	F Δ {+ -}	+を指定すると、これ以降構文の形式も併せて表示する。 -を指定すると表示しない。

注1) 指定できる変数には次の制限がある。
 (1) 変数の型は、整数型、実数型、文字型及び文字列型(packed array)とする。
 (2) 変数名の長さは40文字以内とする。
 (3) 変数宣言の有効範囲(scope)を守る。
 (4) 変数名はプログラム中で記述した表現形式に拘る。
 注2) Aコマンド、Bコマンドで指定する行番号の行には表示点がなければならぬ。プログラムの最大行番号より大きな番号が指定された場合は、最後の文の次(プログラムの終了位置)が指定されたと思える。

いてまとめている。

2.3.3 表示形式

2.3.3.1 標準出力

PASVID は、プログラムの実行の流れが次節に述べる表示点に達すると標準出力を行う。図3に示すように、標準出力は実行制御の位置(①の部分)と補足説明(②の部分)とから成る。

実行制御の位置は、ソース・プログラムにおいてこれから実行をしようとする部分をカーソル(“=”)で指示する。さらに、プログラム上での前後関係を知るため、その行の前後数行ずつを表示する。

補足説明は、実行しようとする文の構文やif文等の論理式の値を表示する。説明部分はF Δ -コマンドにより表示をやめさせることもできる。

2.3.3.2 表示点と補足説明

PASVID が表示を行う点を表示点と呼び、表示点から次の表示点までを1実行制御単位または1ステップという。代表例について二、三説明する。図2の(1)、(2)、(3)にそれぞれ手続き・関数、代入文、およびif文における表

示点とその補足説明の内容を例示している。

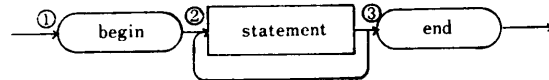
2.4 使用例と課題

二つの整数 m, n の最大公約数を求めるプログラム*(図7参照)に対する使用例を図3に示す。例示したように、プログラムの実行過程がディスプレイ画面上に次々と表示されてゆくの、意図したとおりに動いているかどうかを容易に確かめることができる。そのため初心者ばかりでなく、熟練者のデバッグ・エイドとしても有効である。

PASVID による実行演示の経験を通じて、プログラムのダイナミクスを視覚という観点から解析することにより、プログラムの良否に関してたとえばあちこち目まぐるしく動くプログラムとか整然と動くプログラムというような興味のある結果が得られるのではないかと考えている。

3. 図形作画支援システム

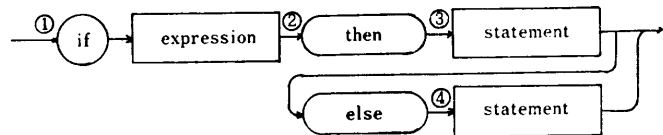
前章の PASVID 等の教具を用いて Pascal の理解が深まると、実行結果を図や表にまとめて視覚的に提示することにより、問題の本質を直観的に分かる



- ① { PROCEDURE } ENTRY: BEGIN <STATEMENT>; ... ; <STATEMENT> END
 - ② 文の種類により異なる。
 - ③ { PROCEDURE } END:
 - ④ { FUNCTION }
- (1) 手続き・関数の場合



- ① ASSIGNMENT STATEMENT: <VARIABLE> := <EXPRESSION>
- (2) 代入文の場合



- ① IF STATEMENT: IF <EXPRESSION> THEN <STATEMENT> ELSE <STATEMENT>
又は
IF <EXPRESSION> THEN <STATEMENT>
 - ② IF EXPRESSION VALUE={ TRUE
FALSE }
 - ③ ④ 文の種類によって異なる。
- (3) if文の場合

図 2 代表的な実行文における表示点と補足説明

Fig. 2 Display points and explanations in some representative statements.

* ACOS-6 Pascal の通常のプログラムは、標準 Pascal の program 文を procedure main に、最後の end. を end に変えればよい。文法の主な相違点は付録に示している。

* PD_GCD.S ... 図7に示す例プログラム(最大公約数の計算)の実行演示

```

PASCAL DEMONSTRATOR                                VER.01.01
-----
00010 PROCEDURE MAIN;(*GCD(GREATEST COMMON DIVIDER) PROGRAM*)
00020   VAR M,N: INTEGER;
00030   BEGIN WRITE(' ');
①      =          ... カーソル(" ")は表示点を示す
00040     READLN(M,N);
00050     WHILE M<>N DO
00060       IF M>N THEN M:=M-N ELSE N:=N-M;
-----
② PROCEDURE ENTRY: BEGIN <STATEMENT>;...<STATEMENT> END

? B #50 ... blindモードで行番号50の先頭まで実行
= 40 16 ... 変数M,Nに値を入力

PASCAL DEMONSTRATOR                                VER.01.01
-----
00020   VAR M,N: INTEGER;
00030   BEGIN WRITE(' ');
00040     READLN(M,N);
00050     WHILE M<>N DO
      =
00060       IF M>N THEN M:=M-N ELSE N:=N-M;
00070     WRITELN('GCD=',M)
00080   END

WHILE STATEMENT: WHILE <EXPRESSION> DO <STATEMENT>

? R+ ... 参照された変数の値をこれ以降表示する
? S ... 1 ステップ実行 以降、画面が変わる度に S コマンドを入力してゆくと
      M,Nの値は { M=40, { M=24, { M=8
                  N=16, { N=16, { N=16
                  と変化し、次の
                  画面に達する。

PASCAL DEMONSTRATOR                                VER.01.01
-----
00020   VAR M,N: INTEGER;
00030   BEGIN WRITE(' ');
00040     READLN(M,N);
00050     WHILE M<>N DO
      =
00060       IF M>N THEN M:=M-N ELSE N:=N-M;
00070     WRITELN('GCD=',M)
00080   END

WHILE EXPRESSION:

? S
M =      8
N =      8

PASCAL DEMONSTRATOR                                VER.01.01
-----
00020   VAR M,N: INTEGER;
00030   BEGIN WRITE(' ');
00040     READLN(M,N);
00050     WHILE M<>N DO
      =
00060       IF M>N THEN M:=M-N ELSE N:=N-M;
00070     WRITELN('GCD=',M)
00080   END

WHILE EXPRESSION VALUE = FALSE ... WHILE文の論理式の値

? B #90 ... blindモードでプログラムの最終位置まで実行

GCD=      8 ... 行番号00070のWRITELN文による結果の出力

PASCAL DEMONSTRATOR TERMINATED. ... PASVIDの終了メッセージ

```

図3 最大公約数プログラムのPASVIDによる実行例
Fig. 3 Example of PASVID for a GCD program.

ように表現したいという要求が発生する。Wilson⁷⁾も指摘しているように Pascal を教育に利用する際の問題点の一つは図形の作画ができないことである。そこ

で、付録にも示しているように ACOS-6[Pascal]に提供されている分割コンパイルおよびライブラリ結合機能⁸⁾を用いて、Pascal の基本理念をできる限り損な

〈作画ライブラリ外部手続き宣言部〉

```

type definitions ;
:
:
procedure plots( formal parameters ) ; extern ;
procedure plot
procedure symbol
procedure scale
procedure line      :      :
procedure number   :      :
procedure axis     :      :
procedure where
procedure factor
procedure newpen
procedure plote( formal parameters ) ; extern ;
    
```

〈作画用プログラム本体〉

```

procedure main ;
:
:
begin
  plots( actual parameters ) ;
  :
  :
  plote( actual parameters )
end
    
```

図 4 Pascal 作画用プログラムの構造
Fig. 4 Structure of Pascal graphic program.

わないよう配慮して、図形作画支援システムを実現した³⁾。

3.1 システムの特徴

3.1.1 作画プログラムの構造

図 4 に Pascal で作画を行うためのプログラムの構造を示す。作画ライブラリ外部手続き宣言部は、プログラム本体で呼び出して使用する作画用ライブラリの宣言を行う部分である。このような構造を採用し、コマンド PDCL (Pascal DeCLarations) を用意してこの宣言部を自動的に生成すれば、その都度学生が入力する煩わしさを除去することができる。利用面での教育的配慮を優先させたため、現時点では標準 Pascal との構文¹⁾の整合性は残るが、コンパイラ・オプションとして標準手続き化されればこの部分は完全になくすことが可能となる。

3.1.2 作画処理

上述した構造のプログラムを翻訳・実行することにより作画が行われる。その際、作画データはディスク・ファイル上に一度書き出されるので、作画機能のない CRT ディスプレイでも実行が可能である。したがって、必要に応じてファイル上の作画データをグラフィック・ディスプレイやプロッタに表示すればよい。また、後述するようにディスク・ファイル上の作画結果をレポートとして計算機内のレポート・ボックスにそのまま提出することもできる。

3.2 作画ライブラリの機能

表 2 Pascal 作画ライブラリ
Table 2 Pascal graphic library.

手続き名・呼出し形式	機能
PLOTS(buff, size)	作画データ用の出力バッファの開設と作画ファイルのオープンを行う。
PLOT(x, y, pen)	座標 (x, y) に対して、位置決め、線描、又は原点の再設定を行う。
SYMBOL(x, y, height, string, angle, nchar)	座標 (x, y) に対して、位置決め又は線描後、文字列の記号を書く。
SCALE(zarray, axislength, npoints, increment)	配列に格納された計算結果の座標データを軸の長さに取めるため、スケリングし初期値と増分値を求める。
LINE(xarray, yarray, npoints, increment, linetype, mark)	配列に格納されている座標データ間を指定されたタイプの線で結び、必要ならばマークも描く。
NUMBER(x, y, height, floating-point number, angle, nmantissa)	座標 (x, y) に位置決め後、浮動小数点数を10進数に変換し、指定された形式で描く。
AXIS(x, y, axisname, nchar, axislength, angle, firstv, deltax)	座標 (x, y) に位置決め後、指定された形式で座標軸及び軸名を書く。
WHERE(x, y, factor)	現時点のペンの位置 (x, y) と図形の倍率係数を知る。
FACTOR(factor)	図形を拡大又は縮小させる倍率係数を与える。
NEWPEN(color)	使用するペンの色を指定する。
PLOTE(x, y)	原点の座標を (x, y) に設定し、更に作画ファイルをクローズする。

表 3 Pascal 作画用コマンド
Table 3 Graphic command for Pascal.

入力形式	機能
PGRUN { ソース・ファイル名 } [; データ・ファイル名]	一時的テキスト・ファイル(*)又はソース・ファイル中のPascal作画プログラムを翻訳・実行し作画を行う。

作画ライブラリは表 2 に示すように基本的な作画機能⁹⁾を実現したもので、これらの機能を基盤にしてより複雑な図形の作画を行うことになる。

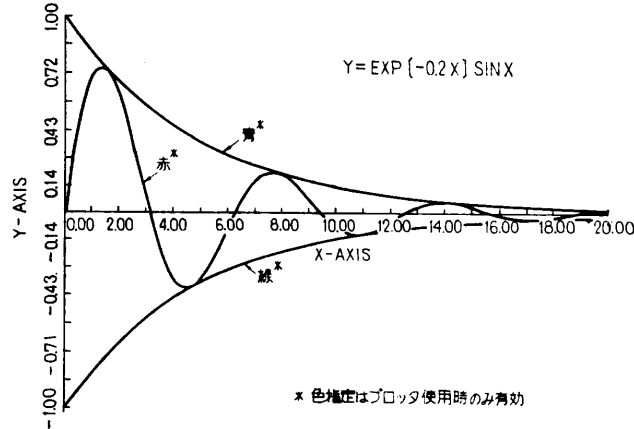
3.3 使用法と使用例

作画用プログラムの翻訳・実行は表 3 に示す PGRUN (Pascal Graphic RUN) コマンドを用いて行う。このコマンドの実行により作画ライブラリとの結合が相対形式プログラムのレベルで行われる。図 5 に $y = ae - bx \sin x$ の作画を行うプログラムと作画結果の例を示す。

4. レポート処理支援システム

4.1 システムの概要

教師が授業で教えた内容に対する学生の理解度を把握するためあるいは種々の問題を課して学生の創意工夫を促進させるためには、レポートは効果的な方法の一つである。しかしながら教師にとっては、レポート・ボックスの用意、提出者や提出日の確認、提出者ごとの仕分け、およびレポートの評価を行うことは結構手間のかかる作業である。このシステムは、情報処理教育等において学生レポートの提出から評価までの



```

*LIST EXPSIN ... グラフィック・プログラム(ファイル名=EXPSIN)のリスト
010 (* ***** Y=EXP(-0.2X)SINX ***** *)
020 (* DECLARATION PART FOR GRAPHIC LIBRARIES *)
030 TYPE IARY=ARRAY[1..20] OF INTEGER;
040 RARY=ARRAY[1..100] OF REAL;
050 PARY=PACKED ARRAY[1..80] OF CHAR;
060 PROCEDURE PLOTS(VAR BUFF:IARY;SIZE:INTEGER);EXTERN;
070 PROCEDURE PLOT(X,Y:REAL;PEN:INTEGER);EXTERN;
080 PROCEDURE SYMBOL(X,Y,HEIGHT:REAL;VAR STRING:PARY;
090 ANGLE:REAL;NCHAR:INTEGER);EXTERN;
100 PROCEDURE AXIS(X,Y:REAL;VAR AXNAME:PARY;NCHAR:INTEGER;
110 AXLEN,ANGLE,FIRSTV,DELTA:REAL);EXTERN;
120 PROCEDURE FACTOR(FACT:REAL);EXTERN;
130 PROCEDURE NEWPEN(COLOR:INTEGER);EXTERN;
140 PROCEDURE PLOTE(X,Y:REAL);EXTERN;
150 PROCEDURE MAIN;(* MAIN BODY FOR GRAPHIC PROGRAM *)
160 CONST N=96;
170 VAR I:INTEGER;
180 BUFF:IARY;
190 W,X,Y,Z:RARY;
200 ST:PARY;
210 PROCEDURE CURVE(VAR X,Y:RARY;N:INTEGER);
220 BEGIN PLOT(X[1],Y[1],3);
230 FOR I:=2 TO N DO PLOT(X[I],Y[I],2)
240 END;(* END OF CURVE *)
250 BEGIN PLOTS(BUFF,20);
260 FACTOR(0.7);PLOT(2.0,10.0,-3);
270 FOR I:=1 TO N DO
280 BEGIN
290 X[I]:=(I-1)*3.141592/15.0;
300 Z[I]:=7.0*EXP(-X[I]*0.2);W[I]:=-Z[I];
310 Y[I]:=Z[I]*SIN(X[I])
320 END;
330 ST:='X-AXIS';AXIS(0.0,0.0,ST,-6,20.0,0.0,0.0,1.0);
340 ST:='Y-AXIS';AXIS(0.0,-7.0,ST,6,14.0,90.0,-1.0,0.143);
350 ST:='Y = EXP(-0.2X)SIN X';SYMBOL(10.0,5.0,0.49,ST,0.0,19);
360 NEWPEN(2);CURVE(X,Y,N);
370 NEWPEN(3);CURVE(X,Z,N);
380 NEWPEN(4);CURVE(X,W,N);
390 PLOTE(21.0,0.0)
400 END

```

```

*PGRUN_EXPSIN ... グラフィック・プログラムの翻訳・実行
DISPLAY? (Y/N) YES ... グラフィック・ディスプレイの場合 YES,
CRTディスプレイの場合 NO を入力する.

```

図 5 Pascal による作画例

Fig. 5 Example of a graph drawn by a Pascal program.

一連の処理を支援するためのものである。ここで学生レポートとは、与えられた課題に対して計算機内のファイルに作成された報告書のことであるが、特に Pascal 演習レポートという場合は、Pascal のプログラム、その入力データと出力結果(図形出力を含む)、

および場合によっては作成者の説明・考察部分をまとめてしたものを目指す。

学生が Pascal で一通りの問題が解ける段階に達すると、教師は演習問題を課す。このシステムを用いることによって、学生は演習レポートをあらかじめ計算

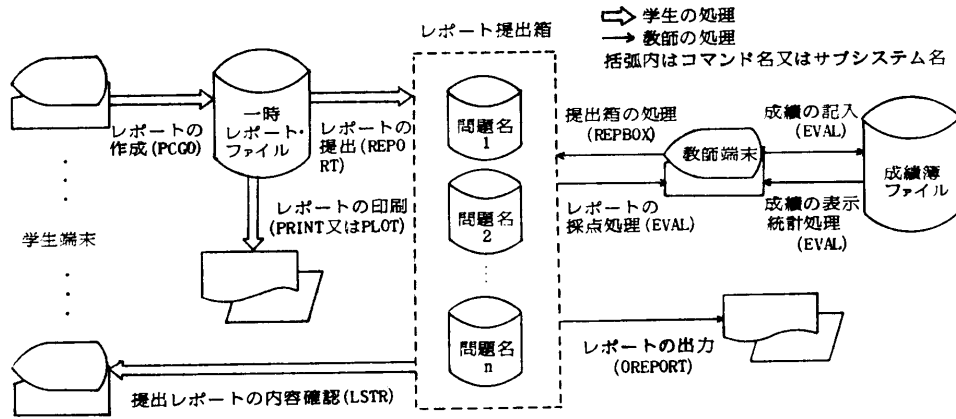


図 6 レポート・ボックス・システムの概念図
Fig. 6 Concept of report box system.

機内に設けられたレポート・ボックスに投函し、教師が必要な時にこれを取り出して評価を行うことができる。図6はシステムの概念図である。計算機内のレポート・ボックスは、ファイル・システムのカatalog構造を利用したもので、箱の用意は問題名と同じ名前をもつカatalogを創成すること、提出はそのカatalogの下にファイルを創成し、レポート内容を複写することに置き換えられる。また、ファイル・システムが提供する保護機能を適用すれば、他人が提出したレポートの自動的な複写や盗み見を防止することが可能となる。

4.2 レポート・ボックスの処理

教師がレポート・ボックスを用意したり、その中の提出者名を表示したりするためのもので表4に示す REPBOX (REPort BOX) コマンドを用いて行う。

4.3 レポートの作成

Pascal 演習レポートの例を図7に示す。TSSにおいてディスプレイ画面による入力・表示が主体の場合には、バッチ方式やハード・コピーがとれる端末の場合と違って、プログラム、入力データ、出力結果等を一まとめにしたこのようなレポートの作成は簡単ではない。そこで、表5に示す PFORM (Pascal FORM), PCGO (PasCal GO) コマンドを用いて、教師に見やすい形で、容易に Pascal 演習レポートを作成する機能を提供する。

4.4 レポートの提出・確認

学生がレポートを提出する場合は、表6に示す REPORT コマンドを用いる。このコマンドにより、図6に示すように教師が前もって用意した同一の問題名をもつレポート・ボックスにその学生専用のレポート・

表 4 レポート・ボックス処理コマンド
Table 4 Command for report box processing.

入力形式	機能
REPBOX { PREPARE OPEN DISPLAY CLOSE RELEASE }, 問題名[一問題名]	ボックスを用意する。 ボックスを開く。 ボックス内の提出者を表示する。 ボックスを閉じる。 ボックスを仕舞う。

*PFORM ... Pascalプログラムの 消書
WAIT A MOMENT.

*PCGO GCD.S;GCD.D ... Pascalレポートの作成
PROBLEM NAME(モンダイメイ)? = EX1 ... 問題名の入力
WAIT A MOMENT.

*REPORT EX1 ... 問題名 EX1 のレポート提出
YOUR REPORT HAS JUST BEEN RECEIVED.

*LSTR EX1 ... 提出したレポートの確認

```
USERID=E552182315 CLASS=034-E PROBLEM=EX1
学生識別名          クラス名          問題名
(ソース・プログラム)          12/17/80 14:49
```

```
00010 PROCEDURE MAIN;(* PROGRAM FOR GCD *)
00020 VAR M,N:INTEGER;
00030 BEGIN WRITE('=');
00040 READLN(M,N);
00050 WHILE M<>N DO
00060 IF M>N THEN M:=M-N ELSE N:=N-M;
00070 WRITELN('GCD=',M)
00080 END
```

(入力データ)

```
80000....1....*....2....*....3....*....4....*
40 16
```

(実行結果)

```
90000....1....*....2....*....3....*....4....*
=GCD= 8
```

図 7 Pascal レポートの提出過程

Fig. 7 Process of preparing a Pascal report.

ファイルが開設され、レポート内容が複写される。もちろん、期限内でないと提出は拒否されるし、提出されたレポートは本人と担当教師だけしか見ることが

MARK OR ADD OR LIST OR STAT OR MODIFY OR DELETE ? L1 ...表示機能の選択
 SCORE(S) OR NO MARK(N) OR REPORT CHECK(R) OR USER ID(U) OR PROB NAMF(P)
 ? S ...成績表の表示
 STUDENT ID ? -T12680 ...学生IDの指定(クラスの先頭の学生からT12680の学生まで)

USER ID (NAME)	問題番号			レポ-ト提出月日 の順位	TOTAL	AVERA	VARIA(MK/ RP)
	1	2	3				
T01992 (ヨシタ)	-	70(4)	-	90(8)	160	80.00	10.00(2/2)
T11956 (イシカワ)	-	30(3)	-	60(7)	90	45.00	15.00(2/2)
T12636 (アノトウ)	-	20(4)	-	70(8)	90	45.00	25.00(2/2)
T12640 (17チ)	60(8)	-	-	80(8)	140	70.00	10.00(2/2)
T12663 (ハハ)	-	-	80(2)	-	80	80.00	0. (1/1)
T12665 (フタミ)	-	-	-	85(7)	85	85.00	0. (1/1)
T12672 (アサリ)	50(8)	-	-	60(8)	110	55.00	5.00(2/2)
T12673 (アハ)	50(7)	30(2)	-	-	80	40.00	10.00(2/2)
T12674 (イケサキ)	50(7)	-	-	80(7)	130	65.00	15.00(2/2)
T12675 (イケナカ)	-	50(3)	-	75(7)	125	62.50	12.50(2/2)
T12676 (ウメ)	90(8)	-	-	65(8)	155	77.50	12.50(2/2)
T12677 (イトウ)	60(7)	-	-	60(7)	120	60.00	0. (2/2)
T12678 (カトウ)	96(8)	-	-	85(8)	181	90.50	5.50(2/2)
T12679 (カワイタ)	88(7)	-	-	*(7)	88	88.00	0. (1/2)
T12680 (カワムラ)	65(7)	-	-	*(7)	65	65.00	0. (1/2)
AVERAGE	65.43	45.83	79.00	73.64		67.23	
VARIATION	19.01	20.90	13.56	10.68		15.69	
NUM OF MARK	21	6	5	11	... 採点した人数		
NUM OF REPT	66	17	8	82	... レポ-ト提出者数		

注) "*" ... レポ-ト未提出
 "*" ... 未採点

図 8 レポ-トの評価例

Fig. 8 Example of report evaluation.

表 5 レポ-ト作成支援コマンド
 Table 5 Supporting commands for preparation of reports.

入力形式	機能
PFORM [△*]	一時的なテキスト・ファイル(*)の Pascal プログラムを見易い形に清書する。
PCGO△ { ソース*ファイル名 } [; データ・ファイル名] [-GRAPH]	一時的なテキスト・ファイル(*)又はソース・ファイル中の Pascal プログラムを翻訳・実行し、プログラム、入力データ及び出力結果をまとめて一時的なレポ-ト・ファイルに書出す。

表 6 レポ-ト・コマンド
 Table 6 Commands for reports.

入力形式	機能
REPORT△ 問題名	指定した問題名のレポ-トを提出する。
LSTR△ 問題名	提出したレポ-トの内容を確認する。
REPORTC△ 問題名	提出済のレポ-トを取消す。ただし前の内容との置き換えはREPORTコマンドを再度実行すればよい。

きない。

提出したレポ-トの確認、取消は、それぞれ、LSTR (LiST Report), REPORTC (REPORT Cancel) コマンドを用いて行うことができる。

4.5 レポ-トの評価

学生が提出した Pascal の演習レポ-トを、教師が必要な時にレポ-ト・ボックスから取り出して評価ができるようにするため、EVAL サブシステム(図6)には次の機能を用意している。

(1) 採点処理機能 この機能は、提出されたレ

ポ-トの内容を教師端末に表示させ、配点項目に応じて評点を入力してゆくもので、その際、レポ-トに書かれたプログラムの再実行やその内容にコメントを付与することもできる。なお、図6に示す OREPORT (Output REPORT) コマンドによりレポ-ト内容を印刷装置に一括出力させ、その評点を成績簿ファイルに記入することも可能である。

(2) 提出者の情報や成績表の表示機能 提出者の氏名や提出日または成績表の内容を端末や印刷装置に出力し、クラスの個々の学生の理解度を把握することができる。図8に使用例を示す。

(3) 成績の統計処理機能 各問題ごとの平均値、標準偏差あるいは度数分布といった統計量の表示を行うことにより、教師はクラス全体の理解度や出題した問題の適切さ等を判断することができる。

4.6 課題

このシステムは、Pascal の演習レポ-トの処理ばかりでなく、計算機を使用した試験や教育相談等にも適用できる。レポ-ト処理のみについて述べるなら、教師にも学生にも好評であるが、考察等をカタカナで入力しなければならぬことおよび文字出力と図形出力が同時に見られないこと等今後改善すべき点も多い。これらは、日本語文書処理技術を導入することに

より解決してゆくことができると思われる。

5. おわりに

本論文では、言語 Pascal の理解を深め、適用範囲を広げ、学生の理解度を把握するための一連の教具について述べた。教具は分かりやすく、使いやすく設計することは当然であるが、実際の教育現場での使用経験が特に重要である。

我々は、教育性に優れた言語 Pascal を通じて学生の情報処理に対する理解をさらに深めるため、現在、教具として Pascal を親言語とする教育用データベース・システムを試作中である。

謝辞 日ごろご指導と有益なご助言を頂いている九州大学工学部牛島和夫教授、大槻説平助教授、並びに長沢勲講師に謝意を表します。

参 考 文 献

- 1) Jensen, K. and Wirth, N.: Pascal User Manual and Report 2nd Edition, Springer-Verlag, New York (1978).

- 2) 宇津宮, 樽美他: プログラム実行演示システムの開発, 信学会総全大会講演論文集, p. 6-130 (1980).
- 3) 荒牧, 宇津宮他: 言語 Pascal のための幾つかの教具, 信学技報 ET 79-4, pp. 17-20 (1979).
- 4) 荒牧, 宇津宮他: 学生レポート提出・評価サブシステム, 九大工学集報, 第52巻, 第1号, pp. 73-77 (1979).
- 5) Kline, R. B., Hamor, G. D. and Druffe, L. E.: Visual Demonstration of Program Execution, SIGCSE Bulletin, Vol. 10, No. 1, pp. 16-18 (1978).
- 6) 藤村, 牛島: FORTRAN プログラム動的解析システムとプリコンパイラ, 情報処理, Vol. 17, No. 6, pp. 547-550 (1976).
- 7) Wilson, I. R.: Pascal for School and Hobby Use, Software Practice and Experience, Vol. 10, No. 9, pp. 6-10 (1980).
- 8) 日本電気: ACOS-6 Pascal 説明書, FGK 01 (1979).
- 9) 富士通: FACOM OS IV/F 4 PSP 文法書, 64 SP-6100-1 (1977).

(昭和56年1月12日受付)

(昭和56年6月16日採録)

付録 ACOS-6 Pascal における標準 Pascal との主な相違点

```

<program> ::= <proc-func-optional> { <proc-func-optional> }
<proc-func-optional> ::= <constant definition part>
                        <type definition part>
                        <variable declaration part>
                        <procedure and function declaration part>
                        <end-of-file>
<procedure and function declaration part> ::=
    { <procedure declaration>; | <function declaration>; }
<procedure declaration> ::= <procedure heading> <block> |
    <procedure heading> <options>
<function declaration> ::= <function heading> <block> |
    <function heading> <options>
<options> ::= forward; | extern; | extern fortran; | extern pascal;

```