

TSSによる基礎プログラミング教育における 問い合わせ方式 REPORT コマンド

羽賀隆洋・太田義勝
(名古屋大学 情報処理教育センター)

1. まえがき 当教育センターにおいては、全学からの授業利用希望にともない、種々の内容の授業利用が行われている。これに対して、授業担当教員及び受講学生の便宜をはかるべく種々の教育支援システムを開発しているが^{(1),(2)}、ここではTSSによる基礎的プログラミング授業の便宜をはかる目的で作成されたREPORTコマンドについて、その作成動機、特徴、通常のEDIT方式⁽³⁾との比較検討などを述べる。

さて、TSSによる基礎的プログラミング授業においては、少なくとも次の3項目を教えることが必須である：(1)プログラミング言語の文法、(2)アルゴリズムの作成法とこのプログラミング言語による実現、(3)TSSによるプログラムの実行法。どのようなプログラミング言語を用いるにせよ、始めて計算機を利用する学生を対象とした、半期の講義においては、上記の3項目を一通り教育するのみでも時間的に相当厳しいことが我々の授業経験から明らかとなった。REPORTコマンドは、上記(3)のTSSの使用法に対する授業時間の負担軽減を主目的として、教育支援の一貫として当教育センターにおいて開発されたものであり、計算機システムから端末画面上に表示されてくる簡単な問い合わせに対してYES、NOによって応答することをその基調としている。

さらに、近年、TSSによる情報処理教育が定着したかに見うけられ、従来のバッチ方式に比較して多くの利点を有していることは新ためて言うまでもないが、その反面TSSにも欠点がないわけではない。すなわち、我々の多少の授業経験によれば、初心者にとってTSSの理解しにくさは特に次の点にあることが判明した：

(1) データの会話的入力法、(2) レポート（すなわち、プログラムの実行結果のリスト）の作成、出力。REPORTコマンドはこれらの欠点を取り除くべく、バッチの入力法の取り入れ、ハードコピー出力法に対する工夫などがなされている。これによって、学生はレポートをTSSによって簡単に作成することが可能となる。

REPORTコマンドの詳細について述べる前に、2.においてまず、当教育センターの教育支援システム開発の方針を述べ、その方針中におけるREPORTコマンドの位置づけを行う。そして、REPORTコマンドの機能などを述べた後に、6.において実際の授業におけるREPORTコマンド使用の効果について検討する。

2. 教育支援システム開発の方針 基礎的プログラミング授業に対する教育支援開発に限って言えば、これらは次の3種に大別できよう。

- (1) 授業担当者の学生に対する指導の援助。
- (2) 学生の自習・体得に対する援助。
- (3) 担当者と学生の間の質問・助言などの対話の援助。

各々の内容を具体的に記せば次のごとくとなる：

- (1) ① 授業優先利用のための端末の確保、解放、② リソースの使用状況を把握するための各種統計出力、③ 学生の出席状況一覧の出力、④ 授業学生全体に対するレポート作成状況の把握及びテストデータによる全学生のプログラムの実行検査、

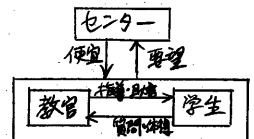


図1. 教育支援システムの一観
Fig.1 One aspect of the education assisting systems.

⑤教官の端末画面の授講学生端末への送信による、TSS使用法教育の援助、⑥学生の端末画面のモニタリング、等。

(2)①学生のTSS利用法学習の負担を軽減し、本来のプログラミング学習に専念し易くさせるための、TSS使用法の簡便化、②すでに用意されているいくつかのプログラムメニューを適宜選択し実行させることにより、学生のプログラミングへの理解を容易にし、又、言語の文法などの自習・復習を可能とさせるシステム、③言語の文法学習の負担を軽減し、又、速いた答を与える教育用簡易言語の開発等。

(3)①端末画面を通じての、教官・学生間の対話の援助、②プログラム相談、プログラミング助言などを取り入れたICAIシステム、等。

以上のうち、(1)⑤、⑥、(3)①の端末画面の送受信体系は現在開発中であり、(3)②は近い将来に実現すべく検討を開始している。その他の大部分はすでに実現され、実用に供されている^{(1),(2)}。

本稿のREPORTコマンドは、(2)①の1つの具体化である。

3. REPORTコマンドの基本方針

3.1 通常のTSS使用法の欠点 我々の授業経験から知られた通常のEDIT方式によるTSSの欠点を述べることにより、REPORTコマンド開発の動機とする。その欠点として知られたのは、以下のごとくである(使用計算機はFACOM M-180IIAD)：

(i) TSSの3つのモード(READY, EDIT, INPUTモード)を常に把握している必要があり、コマンド入力モードでサブコマンドを入力したり、逆に、サブコマンドモードでコマンドを入力する誤まりが頻発する。

(ii) プログラムなどを作成、保存する際にユーザが付ける名前と、それは対応して内部で作られるデータセット名(「登録番号.名前.データセットタイプ」)との対応が理解し難い。特に、データセットタイプ(FORT, FORT77, DATAなど)の概念が把握できない。

(iii) READ命令に対応した、会話的データ入力法が理解し難く、又、データ量が多い時入力困難である。さらには、データの入力誤まりによるエラーに対しては、データ入力の履歴が残されないため異常原因の推定のみにとどまる。

(iv) TSSの進め方が不明となる。例えば、EDITモードを終了してREADYモードへ復帰する方法がわからない。

(v) レポートを簡単に作れない。又、ハードコピー装置を誰か1人が確保すると、解放されるまで他の人は誰もそのハードコピー装置を使用できない。

3.2 REPORTコマンドの特徴 上記3.1のことがらをふまえて、REPORTコマンドの基本方針を次のように定める：

- (i) 問い合わせ方式とする。
- (ii) バッチ方式の長所を取り入れる。
- (iii) レポート作成機能を取り入れる。

以下に、これらの方針を具体的に説明する。なお、REPORTの先頭において使用言語を、表示されたメニューから選択することになっており、現在、FORTRAN, FORTRAN77, FANT(主記憶常驻・Compile and GO形のFORTRAN)⁽⁴⁾、BASICが使用可能である。しかし、REPORTの使用言語による本質的な差異は無いので、本稿では

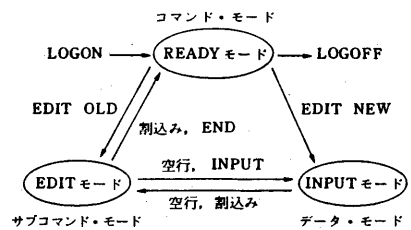


図2. TSSにおける3つのモード
Fig.2 Three modes in TSS.

FORTRANを使用する場合のみを念頭に置いて述べる。

(i) 計算機システムから端末画面に表示される簡単な問い合わせに対して、基本的にはYES, NOで答えることにより、あらかじめ定められたフローによって進みながらレポートを作成できる。従って、REPORT中ではTBSなどのモードにいるかを意識しなくてもよく、又、コマンドの学習もTBSセッション開始、終了用のLOGON, LOGOFF及びデータセット管理用の若干の必要最少限のコマンドを記憶するのみでよい。

(ii) データはプログラムと同一のデータセットに作成しておき、データ部が複数の場合も許される。従って、データのみを変えて実行した結果を比較したりできる。

(iii) プログラム、データ及び実行結果のリストをまとめてレポート形式にする機能をもち、レポートがきわめて容易に作成できる。又、そのレポートのハードコピー装置への出力も、バッチと同様な待ち合わせ形である。

4. REPORTコマンドの実現とその動作 REPORTコマンドは約450行からなるコマンドプロシジャ(TBSのコマンドを定義/作成するための言語)として実現されており、その動作、使用例などについて以下に述べる。

4.1 REPORT中での動作 READY状態においてREPORTコマンドを入力すれば、以下、図3の流れに従って、問い合わせ方式でレポートを作成することができる。REPORTが終了すれば(又は、アテンション入力により強制終了させれば)READY状態へ復帰する。

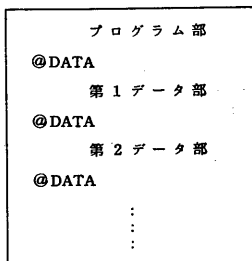


図4. テキストの形式
Fig. 4 Text format.

テキストは図4のように作成、編集しておく。ただし、データ部は不要ならば存在しなくてもよく、又、複数個存在してもよい。なお、データ部が無い場合は、DATA NUMBER? の問い合わせは行われない。図3において、応答が記入されていない場合は、無条件推移、又は、ファンクションキー入力による推移である。

テキストの編集時及び実行結果レポートの表示画面はFS(Full Screen)形式⁽³⁾であり、これらのFS形式画面からの終了は指定されたファンクションキーによる。実行結果レポート画面は見るのみで編集は不可能である。

実行結果レポートのハードコピー出力は、PRINT? に対する応答がYESであればあらかじめ定められた特定の装置へ出力され、応答が検査の場合はその指定された装置へ出力される(いずれも待ち合わせ形である)。

SAVE? に対する応答がYESであればテキストは保存されるが、そのとき内部で作られるデータセット名は「登録番号.テキスト名」となる。

このように、図3のREPORT中での動作フローのみを学生に教えておけば、学生はTBSのコマンド、サブコマンドをほとんど知ること無く、レポートを容易に作成、出力することができる(レポート出力例は、次の4.2の図5参照)。

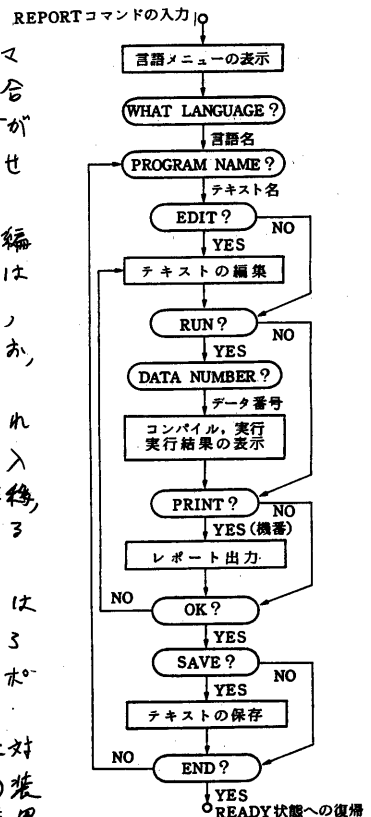


図3. REPORT中での動作
Fig. 3 The flow in the REPORT.

4.2 REPORTによるレポート出力例

REPORTコマンドを使用して作成したレポートの具体例を図5に示す。実行させないとき(RUN?)に対する応答がNOのとき)には、データ部、実行結果部はレポートに含まれない。

なお、図5のプログラムの、次の4.3のREPORT応答時間の測定に用いられる。

さらに、図5はFORTRANの場合のみであるが、他の言語の場合も同様であるので省略する。

4.3 REPORTの応答時間測定

REPORTにおいては、必要な作業用データセットの割付け、編集などのため、若干応答時間が長くなることがある。応答時間は、もちろん、状況によって相当変化する。表1に、図5のプログラムに対する応答時間の測定結果を示す。

T_1, T_4, T_9 以外は言語種類にはほぼ無関係、又、 T_4 以外はプログラム内容にはほぼ無関係のはずであるが、若干の差違が生じているのは状況が刻々変化しているためであると考えられる。Total時間は、大略、簡単な演習問題を行うときのオーバヘッド時間とみなされる。 T_1, T_4, T_9 が特に大であるのは、先に述べたように、必要な作業用データセット割付け、編集などのためであると考えられる。

なお、測定に用いられた計算機はFACOM M-150IID (2.5MIPS)であり、当教育センターには約60台のキヤウクタディスプレイが設置されている。

表1. REPORT 応答時間の測定結果
(各状況において、入手により、数回~10回測定した平均値を示す(秒)。各T_iは箱に示す、各応答入力から応答までの時間である(秒)。(バッチ使用者は、0~3人程度であった)

(1) 素数プログラムの場合

使用端末	ほぼ0台				20~30台程度				40~50台程度				
	FORTRAN	FORTRAN?	PAST	BASIC	FORTRAN	FORTRAN?	PAST	BASIC	FORTRAN	FORTRAN?	PAST	BASIC	
T ₀	1	1	1	1	2	2	2	2	3	3	3	2	3
T ₁	12	11	10	10	18	17	18	18	32	37	28	32	
T ₂	0	0	0	0	0	0	0	0	1	2	1	1	
T ₃	3	3	6	5	5	4	8	8	9	9	9	8	
T ₄	12	12	10	14	25	21	15	20	36	53	13	21	
T ₅	0	0	0	0	0	0	0	0	2	2	2	3	
T ₆	0	0	0	0	0	0	0	0	1	1	1	1	
T ₇	0	0	0	0	0	0	0	0	1	1	1	1	
T ₈	0	0	0	0	0	0	0	0	1	1	1	1	
T ₉	7	6	4	4	10	9	8	10	19	32	15	17	
Total	35	33	31	34	60	53	51	58	105	141	73	88	

(2) ソートプログラムの場合

使用端末	ほぼ0台				20~30台程度				40~50台程度			
	FORTRAN	FORTRAN?	PAST	BASIC	FORTRAN	FORTRAN?	PAST	BASIC	FORTRAN	FORTRAN?	PAST	BASIC
T ₀	1	1	1	1	2	2	2	2	2	2	2	3
T ₁	12	11	10	10	14	14	15	22	33	41	27	25
T ₂	0	0	0	0	0	0	0	0	1	1	1	1
T ₃	6	6	3	4	8	17	8	9	13	13	9	11
T ₄	12	12	8	7	24	25	14	12	37	49	15	21
T ₅	0	0	0	0	1	1	1	1	1	3	2	2
T ₆	0	0	0	0	0	0	0	0	0	0	0	0
T ₇	0	0	0	0	0	0	0	0	0	0	0	0
T ₈	0	0	0	0	0	0	0	0	0	0	0	0
T ₉	6	6	4	4	12	17	8	13	27	26	16	19
Total	37	36	26	25	61	81	48	59	114	135	72	82

```

((MSG))
USER-ID      : ZZ06999
PROGRAM NAME : SOSU
((SOURCE LIST))
FACOM DSIV/F4 FORTRAN IV (GE) U04L11

C****REPORT SYORI-TIME SKUTEIYO.
C****SOSU (DATA WEST)
DIMENSION IP(1000)
NP=0
DO 1000 I=3,999.2
000004  I1=I
000005  DO 1100 J=2,I1
000006  I2=I/J
000007  IF(I2.LE.0) GO TO 1000
000008  1100 CONTINUE
000009  NP=NP+1
000010  IP(NP)=I
000011  1000 CONTINUE
000012  WRITE(6,100) (IP(I),I=1,NP)
000013  100 FORMAT(1H ,10I5)
000014  STOP
000015  END
((DATA))
((RUN))
5 5 7 11 13 17 19 23 29 31
37 41 43 47 53 59 61 67 71 73
79 83 89 97 101 103 107 109 113 127
131 137 139 149 151 157 163 167 173 179
181 191 193 197 199 211 223 227 229 233
239 241 251 257 263 269 271 277 281 283
293 307 311 313 317 331 337 347 349 353
359 367 373 379 383 389 397 401 409 419
421 431 433 439 443 449 457 461 463 467
479 487 491 493 503 509 521 523 541 547
557 563 569 571 577 587 593 599 601 607
613 617 619 631 641 643 647 653 659 661
673 677 683 691 701 709 719 727 733 739
743 751 757 761 769 773 787 797 809 811
821 823 827 829 839 853 857 859 863 877
881 883 887 907 911 919 929 937 941 947
953 967 971 977 983 991 997

((INFORMATION))
DATE (MM/DD/YY) : 04/19/82
REPORT START TIME : 16:33:04
EXECUTION TIME : 9 SEC.
    
```

```

((MSG))
USER-ID      : ZZ06999
PROGRAM NAME : SORT
((SOURCE LIST))
FACOM DSIV/F4 FORTRAN IV (GE) U04L11

C****REPORT SYORI-TIME SKUTEIYO.
C****SORT (DATA ART)
DIMENSION ID(100)
READ(S,100) ND
100 FORMAT(15)
110 READ(S,110) (ID(I),I=1,ND)
110 FORMAT(10I5)
11=ND-1
DO 1000 I=1,11
000007  DO 1100 J=J,ND
000008  IF(ID(I).LE.ID(J)) GO TO 1100
000009  I2=ID(I)
000010  ID(I)=ID(J)
000011  ID(J)=I2
000012  J=J+1
000013  1100 CONTINUE
000014  1000 CONTINUE
000015  WRITE(6,120) (ID(I),I=1,ND)
000016  120 FORMAT(1H ,10I5)
000017  STOP
000018  END
((DATA))
((RUN))
25 25
27 24 23 22 21 20 19 18 17 16
12 14 13 12 11 10 9 8 7 6
5 4 3 2 1
((INFORMATION))
DATE (MM/DD/YY) : 04/19/82
REPORT START TIME : 16:41:56
EXECUTION TIME : 7 SEC.
    
```

(2) ソートの例 (DATAあり)

図5. レポート出力例

Fig.5 Examples of report output.

- T₀: REPORT?—WHAT LANGUAGE?
- T₁: 素数名?—PROGRAM NAME? (データセット割付など)
- T₂: プログラム名?—EDIT?
- T₃: NO?—RUN?
- T₄: YES?—(DATA NUMBER? → I) → 結果表示開始 (コンパイル実行)
- T₅: PF15?—PRINT?
- T₆: NO?—OK?
- T₇: YES?—SAVE?
- T₈: NO?—END?
- T₉: YES?—READY (データセットの削除など)

測定時間のバラつきは、個々の T_i において表中の平均値の高々5割程度の増減、Total時間においては表中のTotal時間の高々2割程度の増減であった。

5. REPORT ジェネレータ 図3のフロー図は教育センターで用意した標準のものであるが、担当教官によっては別の流れ(フローと呼ぶ)により動作させたい要求が生じる。従って、フローの定義を与えて、それに基いて動作するコマンド・プロシジャをシミュレートするシステムを教育センターで用意しておくことが考えられる。但し、必要性などを考慮して、(i)言語はFORTRANとする、(ii)プログラマ部とデータ部を別のデータセットに作成する、場合のみを考へる。

5.1 フローの定義 フローの定義は、次の2種の定義文を用いて行う。ここに、
 { 節点内容定義文 $n \llcorner$ 節点内容文字列 \triangleright n, m は節点番号であり、01, ..., 99の10進2桁の数字とする。推移定義文において、 m は推移先の節点番号を表わし、推移条件は利用者の応答入力を表わす(表2参照)。
 フロー定義における制約:

- (1) 節点内容定義をすべて行ってから、節点間推移定義を行うこと(各々の間では、定義順序は任意でよい)。
- (2) 異なる節点番号で同一節点内容の節点が、複数個現れる。但し、ENTRY(入口)とEXIT(出口)は、各々フローの先頭、終端に丁度1個ずつ与えるものとする。
- (3) 処理、表示節点へは、適当な応答がなされた後に到達すること。

表2. 節点内容、及び、可能な応答

節点内容	可能な応答
ENTRY	=⇒ ①
EXIT	無し
PROGRAM-NAME ?	&PNAME.
DATA-NAME ?	&DNAME. 空行 ²⁾
EDIT-PROGRAM ?	YES, NO
EDIT-DATA ?	YES, NO
RUN ?	YES, NO
PRINT ?	YES(又は、&KIBAN) ³⁾ ; NO
OK ?	YES, NO
NEXT-PROGRAM ?	YES, NO
NEXT-DATA ?	YES, NO
SAVE-PROGRAM ?	YES, NO
SAVE-DATA ?	YES, NO
END ?	YES, NO
EDIT-PROGRAM	=⇒ ④
EDIT-DATA	=⇒ ④
RUN	=⇒
MAKE-REPORT	=⇒
DISPLAY	=⇒ ④
PRINT	=⇒ ⑤
SAVE-PROGRAM	=⇒
SAVE-DATA	=⇒

表3. 処理、表示節点へ到達するまでの必要な応答とエラー時の処置

節点内容	到達するまでの必要な応答	エラー時の処置
EDIT-PROGRAM	プログラム名	EXITへ推移
EDIT-DATA	データ名	警告のみで次へ推移 ¹⁾
RUN	プログラム名	EXITへ推移 ²⁾
MAKE-REPORT	プログラム名	EXITへ推移
DISPLAY	MAKE-REPORTを通過していること	EXITへ推移
PRINT	MAKE-REPORTを通過していること	EXITへ推移
SAVE-PROGRAM	プログラム名	EXITへ推移
SAVE-DATA	データ名	警告のみで次へ推移 ¹⁾

- 1) データを必要としないプログラムがあることに注意。
- 2) データを必要とするプログラムで、データ名が応答入力されていないと、RUN節点実行時にエラーとなり、READY状態へ復帰する。
- ③) フロー定義における節点未定義、節点未使用、推移未定義は、レポート使用時のエラーとなる。推移の重複定義は、最後の定義が有効となる。

- 1) =⇒ は無条件推移を表す。
- 2) この空行入力、DATA部が無いことを表す。
- 3) PRINT ? に対する応答YES、&KIBANのジェネレートへの効果は全く同一であり、どちらか一方のみに対する推移を定義しておけばよい。
- 4) この場合の=⇒は、実際には、ファンクション・キー入力に対応する。
- 5) 出力待ち行列に並んで、推移は先へ進む。

5.2 ジェネレータ フロー定義は教官などのユーザが、5.1の方法で与える。各節点内容に対応するコマンド・プロシジャの原形は、表2の各節点内容に応じてあらかじめ用意されており、その原形においては応答に対する推移先は未定義のままに残されている。又、各原形の先頭には固有のラベルが各々付けられている。

ジェネレータは、まず、フロー定義の前半に現われる節点内容のコマンド・プロシジャの原形をすべて集め、一本のファイルとする。この際、同一内容の節点が2個以上現われれば、対応するコマンド・プロシジャ原形の先頭のラベルを固有のラベルにモディファイして、必要個数の原形を各々複製して含めておく。

次に、上記で作られたコマンド・プロシジャの原形において、フロー定義の後半に現われる推移定義に合わせて、未定義のままに残されていた推移先を各々書き換える。これで、ジェネレートが完了することになる。

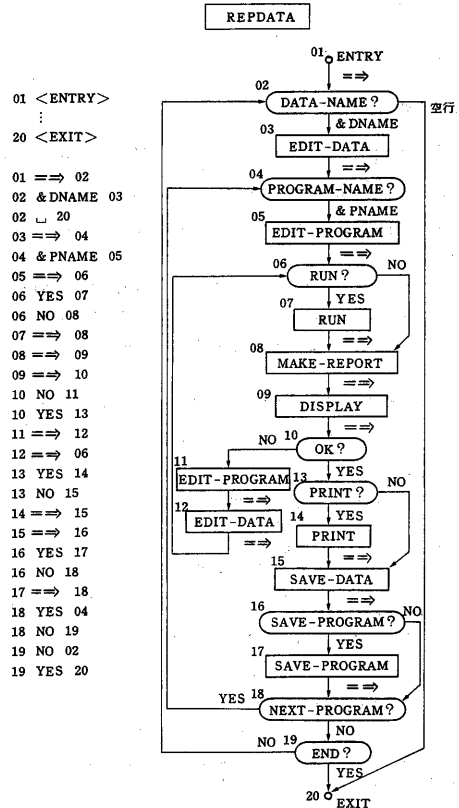
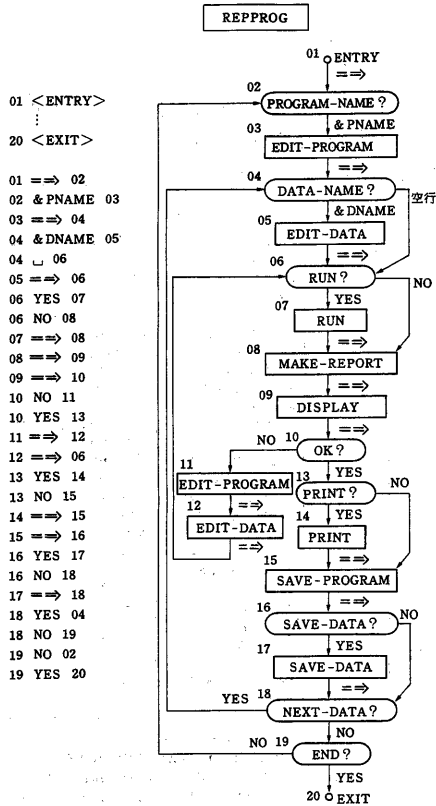
なお、各節点間における情報の受け渡し（例えば、プログラム名が与えられているかどうか、プログラムが実行されたかどうか、レポート形式がすでに作られているかどうか、など）のため、数値のフラグ用変数が用いられている。これは、又、エラー処理のためにも用いられている（表を参照）。

このように、シエネレートはきわめて簡潔なものであるので、これ以上の説明は略す。

5.3. シエネレート例

(1) プログラムを一定にして、データを種々に変えて実行するためのフロー。

(2) データを一定にして、プログラムを種々に変えて実行するためのフロー。



この REPPROG, REPDATA は誰でも使用可能となっており、教官、センター職員などにより実際に使用されている。又、シエネレート時間は、この例において、各々高々 2~3 秒程度であった。

その他、種々のフローを工夫することにより、色々な応用が可能である。

6. REPORT コマンド使用の効果 先の 4. で述べられた REPORT コマンドを実際の授業で使用し、通常の EDIT 方式と比較するための若干の実験を行った。なお、いずれの実験も、フォートランの説明及び T 言語の使用法の基礎を各々 3 回程講義した後に行われた。

[実験 1] 教養部文科系 1 年の授業において、(i) EDIT 方式を用いる、(ii) REPORT 方式を用いる、(iii) EDIT, REPORT の任意の一方を選んで用いる、という 3 グループに分け、プログラムの完成度、所要時間の比較などを行った。なお、制限時間は

20分であり、整数Nを讀込んでプリントし、-1000以下の値が入力されたら停止するというプログラム(7行)をおさめ示して、プログラム入力編集、実行、レポート出力を行せたものである。

この表4より、次のことが知られる:

(1) REPORT方式の方が完成度は高い。特に、第(ii)グループでREPORT方式を選んだ学生に対して、それが顕著である。

(2) REPORT方式の方が、所要時間はわかる。

【実験2】工学部3年の授業において、各学生が通常のEDIT方式及びREPORT方式の両者により、与えられた同一問題(何人かの身長を讀込んで、Xの最大、最小、及び、165cm未満と以上の人数を求める問題)を解くように指示し、2週間程度の後には、比較意見を記入の上レポートを提出させた。Xの結果を表5に示す。

表5より、EDIT方式よりREPORT方式に好意的な学生が多いことが知られる。又、REPORTコマンドの作成意図が十分に良く理解されていたことが、比較意見から知られた。但し、EDITに好意的な学生は断定的に述べているのに対して、REPORTに好意的な学生は部分的肯定が多いという特徴が見うけられたことは注意すべき点である。

さらに、数問題を学期末にレポートで提出させた所(EDIT、REPORTの任意のものを用いて良いと指示)、提出者77人のうち、EDIT方式が22人、REPORT方式が55人であった。

以上の実験結果より、REPORTコマンドは目的とする効果を有し、意図した特徴が実際に現われていることが確かめられたといえる。

7. おすが REPORTコマンドは我々が担当している授業及び他の教官の2,3の授業において実際に利用され、意図した効果をおげている(6.参照)。しかし、REPORT方式がどの授業でも通用するとは限らず、次のような問題提起が予想される:

(1) REPORT方式のみを使用した学生が将来一般社会へ出たとき、通常の計算機使用法と大きく異なっていることを知って困難に直面することはないか。

(2) データ部をプログラムと同一データセットにあらかじめ作成しておくというバッチ方式の取り入れは、会話的に入力するというTJの利点に反していないか。

これらの問題提起に対しては、我々は次のように考えている:

(1) REPORTコマンドは基礎的プログラミング授業の、しかも、プログラミングの理解に手間どっている学生に供するべく開発したものであり、REPORTコマンドの使用により多少とも計算機に対する感覚ができれば、何も理解できずに終るよりはるかに好ましいであろう。又、それにより、より高度な段階へと進められ

表4. 実験1の結果

	学生数	プログラム作成	実行中	プリント出力	完成	
		中時間切れ	時間切れ	時間切れ		
(i) EDIT	27人	8人	2人	1人	16人 (59%)	
(ii) REPORT	32人	12人	0人	0人	20人 (63%)	
(iii) 任意	EDIT	17人	3人	5人	0人	9人 (53%)
	REPORT	4人	0人	0人	1人	3人 (75%)

	完成者の所要時間				完成者・所要時間平均	
	5分未満	5~10分	10~15分	15~20分		
(i) EDIT	2人	7人	1人	6人	11分	
(ii) REPORT	1人	3人	5人	11人	16分	
(iii) 任意	EDIT	0人	1人	5人	3人	13分
	REPORT	0人	1人	1人	1人	14分

表5. 実験2の意見集計結果

EDITに好意的	REPORTに好意的	どっちが良いか断定していない	計
9人	23人	53人	85人

る可能性も大と存する。

(2) TSSの会話的入力といえども、その利点が發揮されるのは極く少量のデータ入力の場合であり、かなりの量のデータを入力する場合、又は、データ部を種々変化させて結果を比較したいような場合には、REPORTにおいて取り入れられた方法がより適切であろう。

最後に、多くの学生が集中したときの、REPORTの応答速度の低下の問題が残されている。本文でも述べたように、それは主として種々の作業用データセットの割付け準備、編集作業のために時間を取っているのであり、そのような必要性の少ない教育用言語を新たに設計することなどが考えられる。しかし、それは簡単に実現できることか否かではなく、今後の課題である。

なお、REPORT中での使用言語として、その後、PASCALが追加されたことを付記する。

謝辞： REPORTコマンドの開発、実現に御協力いただいた名古屋大学の関係者諸氏（特に、当教育センターのシステム専門委員会の各委員）、なすびに、富士通及びFHLの関係者諸氏に感謝する。又、REPORT応答時間の測定及びREPORTの使用実験の結果集計に協力していただいた当センター職員に感謝する。

【文 献】

- (1) 太田, 羽賀: "情報処理教育における支援システム(1)", 情報処理学会第24回全国大会 3P-3 (1982.3).
- (2) 羽賀, 太田: "同 上 (2)", 同上 3P-4 (1982.3).
- (3) "TSS利用の手引", 名古屋大学情報処理教育センター教育広報専門委員会 (1981.5).
- (4) "FACOM OSIV/F4 FAST使用手引書", 富士通マニュアル (1978.8).