

# 初心者のためのTSSコマンドの使い方

— FORTRANプログラム実習における —

宮 地 功\*

(昭和 52 年 4 月 30 日 受理)

How to Use TSS Commands for Beginners

— On Practising FORTRAN Programming —

Isao MIYAJI

(Received April 30, 1977)

This paper describes how to use TSS commands (NEAC 3200 in line TSS) easily for beginners. The Commands are hardly used through mistake by the proposed device. FORTRAN programs can be surely processed in a short time. Practice results are given.

## 1 序 論

人は自分たちのために電子計算機がどのような役割を果たすべきかを正しく認識し、適切に利用することがこれからの社会では要請される。特に工学の分野を志向する技術者は、このことをよく認識する必要があると思われる。

現在、工学系の多くの学校では、電子計算機に直接または間接に接触する時間を多く設ける必要からカリキュラムに実習または演習が取り入れられている。本校でも、工学実験などの1部として電子計算機実習が行なわれている。このような実習では、タイムシェアリングシステム（以下TSSと略す）が最も適していることをすでに報告した<sup>1)</sup>。本校のシステムは、1台の電子計算機に接続された5台の端末機を使用して、ほぼ同時にFORTRAN語の学習ができることを目的としたインラインTSSである。これは、通信回線を利用したオンラインTSSとほぼ同等の機能を持っている。また、このシステムでは、利用者はシステムと会話しながら情報をやりとりする。この会話のための言葉は人間側から見ればシステムに下す命令の集まりで、コマンドと呼ばれる。TSSを利用する実習は、コマンドを介して直接に電子計算機を使用しているような実感が得られるた

め、教育効果のあがるシステム利用法である。

本校機械工学科では、TSSを用いて基本的なFORTRAN語を修得させるため、3年生の工作実習の3週間分（3時間/週）をあてている。従来、この9時間程度では、指導者がTSSコマンドの説明を十分できないので、実習者のコマンドに対する理解が乏しかった。それ故、コマンドを十分使いこなせないで、満足な成果が得られていなかった。この実習で効果をあげるためには、コマンドはプログラムを処理する前にあらかじめ使用できるようにしておく必要がある。ここで報告するTSSコマンド使用法によれば、実習者はコマンドについて概略の説明を30分間程度聞けばよい。それだけで、実習者はTSSコマンドの使用順序の流れ図によって大略を把握し、次節に述べる説明の使用例に従って具体的にこなせば、容易にプログラムを処理できる。

このフローチャート形式のTSSコマンド使用法によって、前に報告したFORTRANプログラム<sup>2)</sup>の処理を行なった結果、9時間程度の短い時間でも初心者にも基本的なFORTRAN語を修得させることが可能であることがわかった。以下に実習例もあわせて報告する。

## 2 コマンドの説明

本校機械工学科では、3年生の工作実習の一部として、TSSを用いた電子計算機実習を1人当たり9時間行なってい

\*機械工学科

る。時間が短いので、TSSコマンドがわかりかけた頃に、FORTRANプログラムの練習が十分できないままに、実習が終わってしまうというのが現実であった。この原因の一つとして、コマンドについて説明する時間が短く、コマンドの機能を十分理解していないため、適切に使用することができないことがあげられる。目的によってパラメータが変わるコマンドについては、このことが特に顕著である。その結果、時間を消費する割には効果があがらないで、止むを得ず延長して実習を行ったり、途中で実習を打ち切るとかの方法を取っていた。

このような状況を改善するため、短い授業時間内で効率よくプログラムを処理できる方法を試みた。通常の実習で使用されるコマンドは、用意されている20個のコマンド<sup>3)</sup>のうち、Table 1 に示した10個のコマンドであると考えら

Table 1. Command table.

No.	Command name and Parameters
1	HELLO_035099, mi ya ji-i
2	EDIT_ S, MTS, NEW
	EDIT_ S, MTS
3	FORTRAN
4	START
5	OUTPUT_ S, MTS, NL
6	BYE
7	COPY_ M, Mmi ya j
	COPY_ S, Mmi ya j
8	PURGE_ M, Mmi ya j
9	CINPUT_ S, MTS
10	RENUMBER_ S, MTS

れる。特に1~6までのコマンドがよく使用される。実習時に使用するコマンドの種類と形式を制限すると、コマンドは大きく分けて次の3種類になることがわかる：(1)パラメータを固定したままでよいか、または付ける必要のないコマンド(FORTRAN, START, OUTPUT, BYE, CINPUT, RENUMBER); (2)利用者によってパラメータが変わるコマンド(HELLO, COPY, PURGE); (3)使用目的によってパラメータが変わるコマンド(EDIT, COPY)。(1)に属するコマンドが半数以上の6個あることは、コマンドの取扱いを容易にする。(1)に属するコマンドは、使い方が一定なので意味さえわかれば簡単に取扱える。(2)に属するコマンドについては、使用する前に利用者個人のパラメータを Table 1 または以下のコマンド説明中の形式に示したように書いておけば、結局(1)または(3)に属することになる。(イタリックの文字が利用者ごとに変わる)一番問題となるのは、(3)に属する2個のコマンド(EDIT, COPY)である。しかし、どちらのコマンドもパラメータの変化が Table 1 のように

2種類に過ぎない。従って、使用例を多く示して使用目的が区別できれば、他のコマンドと同じ扱いができる。使用する範囲を限定しさえすれば、このようにコマンドは単純になる。

次に工夫する必要があるのは、使用法および使用順序を明瞭にすることである。コマンドの使用順序を一目瞭然とするために、Fig. 1 に示すような流れ図を作成した。この流れ図によって処理順序の大略を把握し、以下に述べる説明の使用例を見ながら具体的に処理して行けば、確実に容

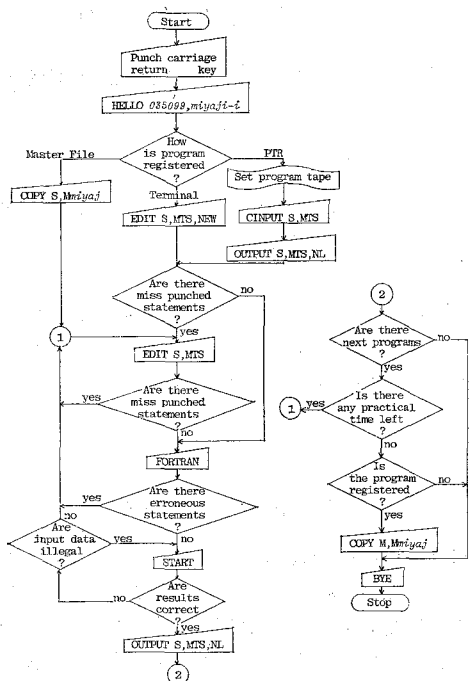


Fig. 1 The order of punching command statements.

易にプログラムを処理できる。

以下に述べるコマンドを説明する手順は次のようである。まず、コマンド名および形式、コマンドおよびパラメータの意味、使用例、次に使用すべきコマンド番号の順に説明する。使用例は、起こると考えられる例を全て挙列した。使用例において、利用者が打鍵する文字は太字で示した。また、⇒に続いて次に使用すべきコマンドを指示したので利用者はこの⇒の指示に従って処理して行けばよい。

0. 会話の開始までの操作

(1)まず、TSS端末タイプライターのプラグをコンセントに入れ、タイプライターの右下にある電源スイッチを LINE (左側)にして端末装置の電源を投入し、オンラインにする。(2)次に、会話要求信号である復帰改行キー "Return" (以下@と略す)を打鍵することによって、システムとの会話を始める。この信号が受け付けられた場合には、シス

テムから次の使用例のような応答がある。\*の次にHELLO  
コマンドを打鍵することになる。

〈使用例〉

Ⓡ ..... 会話要求信号である復帰改行キー Return を入力する。  
WELCOME

READY 12-03-21..... 会話要求信号が受け付けられた時刻  
(時一分一秒)

\*

⇒ 会話を開始するため、1(1)に行く。

1. HELLOコマンド形式 `HELLO_035099, mi ya ji-i`

このコマンドは、システムとの会話を開始させる。第1  
パラメータは利用者番号である。利用者番号は全部で6桁  
あり、各桁の表わ 0 3 5 0 9 9  
す意味は、右のよ (学年) (入学年) (学籍番号の下2桁)  
うになっている。第2パラメータは10文字以内で利用者名  
を入力する。

〈使用例〉

(1)正しくコマンドを入力した場合

READY 12-04-40

\*HELLO\_035099, MIYAJI-I Ⓡ

WAIT 12-06-08

HELLO AT 77/04/25 ..... 受付年月日

NEAC-3200 TSS-32 VER. 0.6 ..... システム名

EOP 00-00-01

⇒プログラムを新しく入力する場合で、端末から入力す  
る時、2(1)に行く。

紙テープ読取装置(PTR)から入力する時、9に行く。

前にファイルに登録してあるプログラムを呼び出して  
使用する場合は、7(2)に行く。

(2)誤ってコマンドを入力した場合

READY 12-51-54

\*HERRO\_035099, MIYAJI-I Ⓡ..... コマンドのつづりの誤り

WAIT 12-52-17

INVALID COMMAND ..... エラーメッセージ

TYPE IN "HELLO" COMMAND

\*HELLO\_035099, MIYAJI-I Ⓡ..... 正しく入力する

WAIT 12-52-36

HELLO AT 77/04/25

NEAC-3200 TSS-32 VER. 0.6

EOP 00-00-01

⇒行先は、1(1)の場合とまったく同じである。

2. EDITコマンド 形式 `(1)EDIT_S, MTS, NEW`  
`(2)EDIT_S, MTS`

このコマンドは、(1)ソースプログラムファイルへの新た  
な登録、および(2)すでに登録されているソースプログラム  
ファイルの内容を修正する。このコマンドの第1パラメー  
タSは、ソースプログラムファイルクラスであることを示  
す。第2パラメータMTSは、作成または修正するファ  
イルの名前がMTSであることを示す。最後のパラメータ  
NEWは、入力情報に各行ごとに行番号を付加しながら、  
ソースプログラムファイルへ新たに登録することを示す。  
(2)の場合は、最後のパラメータNEWは入力しない。修正

は、1行ごとに置換、挿入、および消去できる。ソース  
テキストメントは、1行70文字まで入力できる。

〈使用例〉

(1)ソースプログラムファイルの新たな登録

READY 11-58-09

\*EDIT\_S, MTS, NEW Ⓡ

WAIT 11-58-34

\*00100 \$

C \*\*\* MONDAI 99-(1) 3-1-99 MIYAJI-I Ⓡ

\*00200 \$

C \*\*\* SUICHOKU-KORYOKU NO KEISAN Ⓡ

\*00300 \$

READ(1,5100) F, THETA Ⓡ

\*00400 \$

5100 FORMAT (2F10.0) Ⓡ

\*00500 \$

R=F \* COS. (THETA) Ⓡ

\*00600 \$

WRITE (1,6000) F, THETA, R Ⓡ

\*00700 \$

6000 FORMAT

(1H\_,4X,2HF=,1PE15.7,4H\_(N)/)Ⓡ

\*00800 \$

\* 1H\_,6HTHETA=,E15.7,6H\_(RAD)/Ⓡ

\*00900 \$

\* 1H\_,4X,2HR=, E15.7,4H\_(N)/)Ⓡ

\*01000 \$

STOP Ⓡ

\*01100 \$

END Ⓡ

\*01200 \$

\$ \* Ⓡ

\*01300 \$

1EOF Ⓡ

EOP 00-00-13

⇒修正する行がなければコンパイルするため、3(1)に行  
く。

修正する行があれば、修正のため2(2)に行く。

(2)ソースプログラムファイルの修正(置換、挿入、削除)

①プログラム1からプログラム2への修正例

READY 12-11-49

\*EDIT\_S, MTS Ⓡ

WAIT 12-12-10

\*00100 \$ ..... 修正したい行番号の入力

C \*\*\* MONDAI 99-(2) 3-1-99 MIYAJI-I Ⓡ } 置換

\*00300 \$ ..... 修正したい行番号の入力

1000 READ (1,5100) F, THETA Ⓡ } 置換

\*00450 \$ ..... 挿入したい間の行番号(00401~00499)の入力

IF(F.EQ.0.0) STOP Ⓡ ..... 挿入文(FORMAT文と算術代入文の間) } 挿入

\*01000 \$ ..... 修正したい行番号の入力

GO TO 1000 Ⓡ ..... プログラム1ではSTOP文であった } 置換

\*1EOF Ⓡ ..... 修正が終了することを知らせる

EOP 00-00-06

②プログラム3からプログラム4への修正例

READY 12-19-55

\*EDIT\_S, MTS Ⓡ

WAIT 12-23-14

\*00100 \$ ... 修正したい行番号の入力 } 置換  
**C \*\*\* MONDAI 99(4) 3-1-99 MIYAJI-I** ®  
 \*00350 \$ .....挿入したい間の行番号(00301~00399)の入力 } 挿入  
**VREAC(X,Y)=X\*COS(Y)** ® ... DIMENSION 文と READ 文の間に挿入  
 \*00900 \$ } 置換  
**R(I)=VREAC(F(I),THETA(I))** ®  
 \*1EOF ® .....修正終了を知らせる。  
 EOP 00-00-05

③プログラム4からプログラム5への修正例

READY 12-26-13  
 \*EDIT\_S, MTS ®  
 WAIT 12-26-33  
 \*00100 \$ .....修正したい行番号の入力 } 置換  
**C \*\*\* MONDAI 99(5) 3-1-99 MIYAJI-I** ®  
 \*DELET \$ .....DELETと入力して削除を知らせる } 削除  
**00350** ® .....削除したい行番号の入力  
 \*01810 \$ .....挿入したい間の行番号の入力 }  
**FUNCTION\_VREAC(X,Y)** ® } END行と \$  
 \*01820 \$ } \*の間にこ  
**VREAC=X\*COS(Y)** ® } れらの関数  
 \*01830 \$ } 副プログラ  
**RETURN** ® } ムを挿入す  
 \*01840 \$ } る。  
**END** ®  
 \*1EOF ® .....修正終了を知らせる。  
 EOP 00-00-08

⇒修正する行がなければ、コンパイルするため3(1)に行く。

修正する行があれば、もう一度この2(2)を行なう。

(3)誤ってコマンドを入力した場合

READY 15-02-46  
 \*EJIT\_S, MTS ® .....コマンドのつづりの誤り  
 WAIT 15-03-05  
 ILLEGAL COMMAND  
 EOP 00-00-01

⇒2(1)または2(2)に行つて、正しく入力する。

<注意>

(i)ソースプログラムファイルの内容の修正は、行番号の小さい順に行なわなければならない。

(ii)打鍵した文字の訂正について

①数文字訂正したい場合⇒消去したい文字の数だけ“@”キーを押すことにより、その数だけ文字を消すことができる。

例; \*00500 \$  
**R=F : (a)\*COS@((THTA@)ETA)** ®  
\*のまちがい (の)のまちがい ...HとTの間にEが抜けていた

②1行全部訂正したい場合で、まだ@を打鍵していない場合⇒“←”キーを押すことにより、現在入力中の行を打鍵しなおすことができる。

例; \*00400 \$  
**5100\_FORTRAN(2F10.0)** ← .....つづりのまちがい、  
**5100\_FORMAT(2F10.0)** ®

③1行全部訂正したい場合で、すでに@を打鍵している場合⇒この行より後の登録または修正を全て終え、

1 EOFを打鍵した後、2(2)に行つて修正する。

(iii)メッセージが出力された場合

①SEQUENCE ERROR · RE-TRY

入力した行番号が小さい順になっていない場合出力される。

例; \*01300 \$  
**GO\_TO\_1000** ®

\*00700

SEQUENCE ERROR · RE-TRY

⇒このメッセージの出力された直前の行番号(例では01300である)より大きな行番号の行を全て修正し1EOFを入力する。その後、2(2)に行つて、小さい順になっていなかった行(例では00700である)を修正する。

②DELET NOT FOUND · RE-TRY

消去したい行番号が見つからない場合出力される。

例; \*DELET \$

00350 ®

DELETE NOT FOUND · RE-TRY

⇒同じ EDIT コマンド中に入力した行を消去しようとした場合、他の修正すべき行をすべて修正し、1EOFを入力する。その後、2(2)に行つてこの行(例では00350である)を消去し直す。

前の EDIT コマンド中ですでに消去した場合、すでにこの行(例では00350である)は消去されているので、他の修正を行なえばよい。

③ILLEGAL LINE NUMBER · RE-TRY

行番号、“1EOF”または“DELET”の指定に誤りがあった場合に出力される。

例; \*60000

ILLEGAL LINE NUMBER · RE-TRY

\*1EOF ® .....0がRになっている

ILLEGAL LINE NUMBER · RE-TRY

\*DERET .....LがRになっている

ILLEGAL LINE NUMBER · RE-TRY

⇒再度正しい行番号、“1EOF”または“DELET”を入力する。

3. FORTRAN

形式 FORTRAN

このコマンドは、すでに登録されているソースプログラムファイルの内容をコンパイルする。エラーがなかった場合ライブラリーファイルとリンクロードを行ない、オブジェクトプログラムファイルを作成する。

<使用例>

(1)エラーがなかった場合

READY 12-41-32

\*FORTRAN ®

WAIT 12-41-40

\*LINK LOAD \*... コンパイルエラーがなく、リンクロードの処理を開始したことを示す。

EOP 00-00-20 ..... コンパイルおよびリンクロードの処理に要した時間(20秒)

⇒実行するため、4(1)に行く。

(2)コンパイルエラーがある場合

READY 12-46-08

\*FORTRAN ®

WAIT 12-46-16

3 UNRECOGNIZABLE STATEMENT  
6 MULTIPLE STATEMENT NUMBER  
11 IMPROPER FORMAT ENDING  
12 UNRECOGNIZABLE STATEMENT  
MISsing STATEMENT NUMBER 6000

エラーのある行を示す

ERR FT02

EOP 00-00-09 .....コンパイルに要した時間(9秒)

⇒誤りのある箇所を見つけて、修正のため2(2)に行く。

(3)ロードエラーがある場合

READY 12-59-08

\*FORTRAN ®

WAIT 12-59-18

\*LINK LOAD \*

MISsing, YOMU .....YOMUというサブルーチンがない。

ERR LD04

EOP 00-00-30

⇒誤りのある箇所を見つけて、修正のため2(2)に行く。

(4)コマンドを誤って入力した場合

READY 12-42-44

\*FORTRAN ® .....つづりがまちがっている。(0が0になつている)

WAIT 12-43-05

ILLEGAL COMMAND .....エラーメッセージ

EOP 00-00-01

⇒3(1)に行つて、正しく入力する。

#### 4. STARTコマンド 形式 START

このコマンドは、オブジェクトプログラムの実行を開始させる。

〈使用例〉

(1)データを書式通りに正しく入力した場合

①プログラム1のデータを入力する例

READY 12-42-08

\*START ®

WAIT 12-42-17

# 1.2345 0.5236 ®

F=1.234500E 00 (N)

THETA=5.236000E-01 (RAD)

R=1.069107E 00 (N)

STOP

EOP 00-00-04 .....実行に要した時間(4秒)

②プログラム2のデータを入力する例

READY 12-32-42

\*START ®

WAIT 12-32-56

# 1.2345 0.5236 ®

F=1.234500E 00 (N)

THETA=5.236000E-01 (RAD)

R=1.069107E 00 (N)

# 6.7890 0.7853 ®

F=6.789000E 00 (N)

THETA=7.852999E-01 (RAD)

R=4.801018E 00 (N)

# 9.8760 0.4720 ®

F=9.875999E 00 (N)

THETA=4.719999E-01 (RAD)

R=8.796164E 00 (N)

# 0.0 ®

STOP

EOP 00-00-10 .....実行に要した時間(10秒)

③プログラム3~6のデータを入力する例

READY 12-57-53

\*START ®

WAIT 12-59-31

# 0.0000 3

# 1.2345 0.5236 ®

# 6.7890 0.7854 ®

# 9.8760 0.4720 ®

F=1.234500E 00 (N)

THETA=5.236000E-01 (RAD)

R=1.069107E 00 (N)

F=6.789000E 00 (N)

THETA=7.853999E-01 (RAD)

R=4.800539E 00 (N)

F=9.875999E 00 (N)

THETA=4.719999E-01 (RAD)

R=8.796164E 00 (R)

STOP

EOP 00-00-10 .....実行に要した時間(10秒)

⇒結果が正しい場合、5(1)に行く。

結果が誤っている場合、データの誤りならば、4(1)に行く。

そうでなければ、プログラム修正のために2(2)に行く。

(2)入力するデータの書式が誤っている場合

READY 13-00-58

\*START ®

WAIT 13-01-10

# 1.2345\_0.5236 ® .....入力データが2F10.0の書式になつて

ILL. INPUT DATA .....エラーメッセージ

EOP 00-00-02 .....実行に要した時間(2秒)

⇒4(1)に行つて、データを正しく入力する。

(3)実行警告エラーのある場合

READY 13-20-26

\*START ®

WAIT 13-20-51

# 1.2345 ®

OVFL (DIV) .....除算のオーバーフロー

PAUSE

\*END ® .....実行を終了させる。

EOP 00-00-04 .....実行に要した時間(4秒)

⇒誤りのある箇所を見つけて、修正のため2(2)に行く。

ただし、データの入力誤りならば、4(1)を繰り返す。

(4)コマンドを誤って入力した場合

READY 12-52-42

\*STAT ® .....つづりがまちがっている。

WAIT 12-53-02

ILLEGAL COMMAND .....エラーメッセージ

EOP 00-00-01

⇒4(1)に行つて、正しくコマンドを入力する。

#### 5. OUTPUTコマンド 形式 OUTPUT\_S, MTS, NL

このコマンドは、すでに登録されているソースプログラムファイルの内容を端末に印刷する。第3パラメータNLは、行番号を省略することを意味する。

〈使用例〉

(1)正しくコマンドを入力した場合

READY 12-06-47

```
*OUTPUT_S, MTS, NL Ⓜ
WAIT 12-07-35
TYPE PUNCH SWITCH ON
* Ⓜ .....Ⓜを押した後、印刷されるまでには少し時間がかかる。
C *** MONDAI 99-(1) 3-1-99 MIYAJI-I
C *** SUICHOKU-KORYOKU NO KEISAN
READ (1,5100) F, THETA
5100 FORMAT (2F 10.0)
R=F * COS (THETA)
WRITE (1,6000) F, THETA, R
6000 FORMAT(1HU,4X,2HF=,1PE15.7,4HU(N)/
* 1HU,6HTHETA=,E15.7,6HL(RAD)/
* 1HU,4X,2HR=,E15.7,4HU(N)/
STOP
END
```

```
$ *
IEOF
TYPE PUNCH SWITCH OFF
* Ⓜ .....これを忘れないこと。
EOP 00-00-11
```

⇒今のプログラムを修正して、次のプログラムを処理する場合、2(2)に行く。  
新しいプログラムを入力する場合、2(1)に行く。  
処理するプログラムがなければ、会話を終了させるため、6に行く。

(2)コマンドを誤って入力した場合

```
READY 13-01-36
*OUTPUT_S, MTS, NL Ⓜ .....つづりの誤り(0を0としている)
WAIT 13-02-40
ILLEGAL COMMAND .....エラーメッセージ
EOP 00-00-01
```

⇒5(1)に行って、正しくコマンドを入力する。

6. BYE コマンド

形式 BYE

このコマンドは、会話を終了させる。

〈使用例〉

```
READY 13-27-48
*BYE Ⓜ
WAIT 13-29-33
BYE AT 77/04/25
TIME USED 00-06-26 .....使用したCPUタイム(6分25秒)
```

⇒終わりです。タイプライターの電源スイッチを切り、プラグをコンセントから抜く。

7. COPY コマンド

形式 (1)COPY\_M, Mmiyaj  
(2)COPY\_S, Mmiyaj

このコマンドは、すでに登録されているファイルと同一の内容を他のファイルへ新たに登録する。すなわち、(1)ソースプログラムファイルの内容をマスターファイルに登録するか、(2)マスターファイルに登録してある内容を端末のソースプログラムファイルに移す。第1パラメータには、(1)の場合Mを、(2)の場合Sを指定する。第2パラメータはファイル名でMmiyajなどと頭文字をMとして、続いて利用者名を5文字以内で指定する。この頭文字Mは、機械工学科を表わしている。

〈使用例〉

(1)ソースプログラムファイルの内容をマスターファイルに登録する場合

```
READY 13-10-55
*COPY_M, Mmiyaj Ⓜ
WAIT 13-11-07
EOP 00-00-03
```

⇒会話を終了するため、6に行く。

(2)マスターファイルに登録してある内容を端末のソースプログラムファイルに移す場合

```
READY 12-26-42
*COPY_S, Mmiyaj Ⓜ
WAIT 12-27-00
EOP 00-00-02
```

⇒プログラムを修正するため、2(2)に行く。

(3)指定したファイル名がすでに使用されている場合

```
READY 13-03-40
*COPY_M, Mmiyaj Ⓜ
WAIT 13-04-01
ALREADY FILE NAME
EOP 00-00-01
```

⇒前に登録していたファイルを消去するため、8(1)に行く。

(4)指定したファイル名が登録されていない場合

```
READY 13-04-04
*COPY_S, Mmiyaj Ⓜ
WAIT 13-04-27
ILLEGAL FILE NAME
EOP 00-00-01
```

⇒ファイル名を誤って入力した場合、7(2)に行って正しく入力する。  
プログラムを登録していない場合、登録のため2(1)に行く。

(5)コマンドを誤って入力した場合

```
READY 13-02-43
*COPY_S, Mmiyaj Ⓜ .....つづりが誤っている。(0が0になっ  
WAIT 13-03-38 .....ている)
ILLEGAL COMMAND
EOP 00-00-01
```

⇒7(1)または7(2)に行って、正しく入力する。

8. PURGE コマンド

形式 PURGE\_M, Mmiyaj

このコマンドは、指定したファイルをマスターファイルから消去する。第1パラメータはMとし、第2パラメータは消去したいファイル名を指定する。

〈使用例〉

(1)正しくコマンドを入力した場合

```
READY 12-25-52
*PURGE_M, Mmiyaj Ⓜ
WAIT 12-26-40
EOP 00-00-01
```

⇒プログラムを登録するため、7(1)に行く。

(2)指定したファイル名が登録されていない場合

```
READY 13-04-31
*PURGE_M, Mmiyaj Ⓜ
WAIT 13-05-01
ILLEGAL FILE NAME
EOP 00-00-01
```

⇒ファイル名を誤って入力した場合、8(1)に行って正しく入力する。  
そうでない場合、7(1)に行ってプログラムを登録する。

**9. CINPUT コマンド** 形式 `CINPUT_S,MTS`

このコマンドは、センターの紙テープ読取装置 (PTR) よりソースプログラム紙テープを入力し、各行ごとに行番号を付加しながらソースプログラムファイルに登録する。

〈使用例〉

```
READY 13-05-04
* CINPUT_S, MTS ®
WAIT 13-07-06
EOP 00-00-10
```

⇒プログラムのリストを印刷するため、5(1)に行く。

**10. RENUMBER コマンド** 形式 `RENUMBER_S,MTS`

このコマンドは、すでに登録されているソースプログラムファイルの行番号を付け直す。

〈使用例〉

```
READY 13-10-11
* RENUMBER_S, MTS
WAIT 13-10-31
EOP 00-00-04
```

⇒プログラムを修正するため、2(2)に行く。

**3 実 習 例**

5 台の TSS 端末を用いて、前節に述べたコマンド使用法に従って1回3時間の実習を3回行なった。1回の実習人数は、4~5人である。前に報告した FORTRAN プログラム<sup>2)</sup>の作成および処理を19人の実習者について行なったところ、全員が6個のプログラムを実習時間内に完成できた。その結果を Table 2 にまとめて示す。この表につい

Table 2. The summary of practice results.

Item	Average
No. of jobs for six programs	1.6
No. of commands	36.1
Used time (sec)	519
Used linage	747

て検討する。1個のプログラムを完成するための処理件数は、平均1.6件であった。このことはプログラムの処理が非常に円滑に行なわれていることを示している。また、6個のプログラムを処理するのに必要なコマンド数は、エラーがなければ最低30個必要である。表に示した平均36.1という値は、1個のプログラムを処理するのに約1個のコマンドが余計に使用されただけであることを示している。平均処理件数が1.6件であることから、2個のプログラムを処理するのに約1件の失敗をしていることになる。また、1件余計に処理するには、修正のために EDIT コマンドおよびコンパイルするために FORTRAN コマンドの2つのコマンドが必要である。これらのことから、コマンドはほ

とんど誤りなく使用されていることがわかる。Table 2 の使用した CPU タイムから考えると、1個のプログラムを完成するためには CPU タイムは平均でたったの約86.5秒であることがわかる。また、この実習で1人当たりロール紙を約747行使用しているのがわかる。これは長さにして約3mに相当する。ちなみに、ロール紙1巻の長さは105mであるので、40人のクラスにこの方法で実習しても、ロール紙は1巻と少し消費するに過ぎないことがわかる。

**4 結 言**

ここで提案した方法は、TSSを初めて使用しようとした人が、TSSコマンドを誤りなく使用する方法を示したものである。この方法に従ってコマンドを使用すれば、短時間に確実に FORTRAN プログラムを処理できることが実習結果からもわかる。この方法の特徴をあげれば次のように言える：

- (1) 初歩の FORTRAN プログラムを処理するのに必要なコマンドのみを使用するようにしている；
- (2) あいまいさを少なくするために、コマンドのパラメータを極力固定し、不必要なパラメータを省略している；
- (3) コマンドの使用順序の大略を容易に理解させるため、使用順序の流れ図(Fig.1)で示している；
- (4) コマンドの意味を簡単に説明し、起こると考えられる例を全て示し、更にそれぞれの使用例に続いて次に使用すべきコマンドを指示している。すなわち、フローチャート形式でコマンドの説明をしているのでわかりやすい。

TSSを用いる実習では、このような特徴を持つTSSコマンドの使用方法に従って、前に報告した FORTRAN プログラム<sup>2)</sup>を作成し処理すれば、9時間程度の短い時間でも初心者にも FORTRAN 語の初歩を修得させることが可能である。

最後に、本報告を通読して戴き有益な御教示を戴いた本校岸本俊祐先生、常日頃からいろいろと御指導戴いている本校中原寿喜太先生、また実習の雑事をうまく処理して戴いた赤堀登美子氏の方々に深く感謝致します。

**参 考 文 献**

- 1) 宮地 功；津山高専紀要, No.13 (1975) 11-20.
- 2) 宮地 功；津山高専紀要, No.14 (1976) 65-71.
- 3) 日本電気KK；NEAC3200 TSS-32コマンド説明書, (1973).