島根大学教育学部紀要（自然科学）第18巻　35～43頁　昭和59年12月

情報処理教育のためのコマンドプロシジャについて

福　島　　誠*

Makoto FUKUSHIMA

A Study of Command Procedure for Information Processing Education

I まえがき

あらゆる分野においてコンピュータの利用が常識化しつつある現在において，教育分野においてもコンピュータの利用は CAI のみならず，学校事務の機械化などによって日常化しつつあるといえる。従ってこれからの教員養成においては，コンピュータに関する情報処理教育の必要性が増大してくると予想できる。教員養成大学における情報処理教育は，将来の情報処理技術の専門家をを養成するのが目的ではないので，情報処理教育として適当なのはコンピュータに関する基礎的知識の習得と，プログラミング言語による情報処理実習といったところであると考えられる。このうち後者のプログラミング言語による実習は，実際にコンピュータを使用するということで，単なる言語の知識の習得にとどまらず，体験的な教育効果があるといえる。またこうした情報処理実習における学生の利用状況を調査することは，授業の理解度の確認などの教育的効果の評価に不可欠といえるものである。しかし，実際にはこうした個々の学生の利用状況を把握することは容易ではない。情報処理実習として，大学にある計算センターのシステムを利用する場合，多人数の学生の利用頻度，実行内容等の統計情報を詳細に採取するにはセンター側の協力がかなり必要となる[1]。また，最近の実習の形態としては，以前のようなカード入力によるバッチ形式から TSS 会話処理形式に移行してきているので，JCL (Job Control Language) による一定の処理しか行なわないバッチ処理に比較して，コンピュータと対話しながら処理を行なう TSS 会話処理ではある程度の慣れを必要とし，さらに会話処理によるため，統計情報の採取もバッチ処理に比較すると複雑になる。従って TSS 会話処理による実習を行なうには，事前に端末操作を習得させること，及び実習後の学生の使用状況の把握が問題となってくる。

今回試作したコマンドプロシジャはこうした TSS 会話処理（以下 TSS と略記）実習上の問題点をある程度解決すること目的としたものである。即ち，基本的な TSS 操作が簡単な応答で実行でき，利用状況の統計もとれる機能を有している。またこのコマンドプロシジャは利用者が自由に作成登録可能なので，センターのシステムのオウンコーディングの変更等の，センター側での作業はほとんど必要としない。

II 情報処理教育実習の形態

計算センターのシステムのような大型ホストコンピュータを利用して情報処理教育実習を行う形態としては，前述のように，バッチ処理と TSS の2種類に大別される。ここではこの2種類の利用形態について簡単に比較検討する。バッチ処理の代表的な形態として通常考えられているのがカード入力による方法である[2]。この処理形態は入出力装置としてカードリーダとラインプリンタが最低一台ずつ用意されれば，オフラインのカードパンチ機が複数台必要なだけで，多数のジョブを処理するにはシステムの負担が少なく適しているといえる。しかし，この方法の欠点として，カード利用によるプログラムの編集の困難さがあり，せいぜい100ステップまでのプログラムか，或いはほとんど修正しないプログラム向きの処理形態といえる。初心者ともいえる学生にこうしたバッチ処理による実習をさせることは，設備上の都合による場合は別として，カードの抜き差しによる修正の煩わしさからコンピュータに対するマイナスのイメージを与えかねない。これに対して TSS による実習は1人または2～3人の学生に1台の端末を必要とするため，

* 島根大学教育学部技術科電気工学研究室

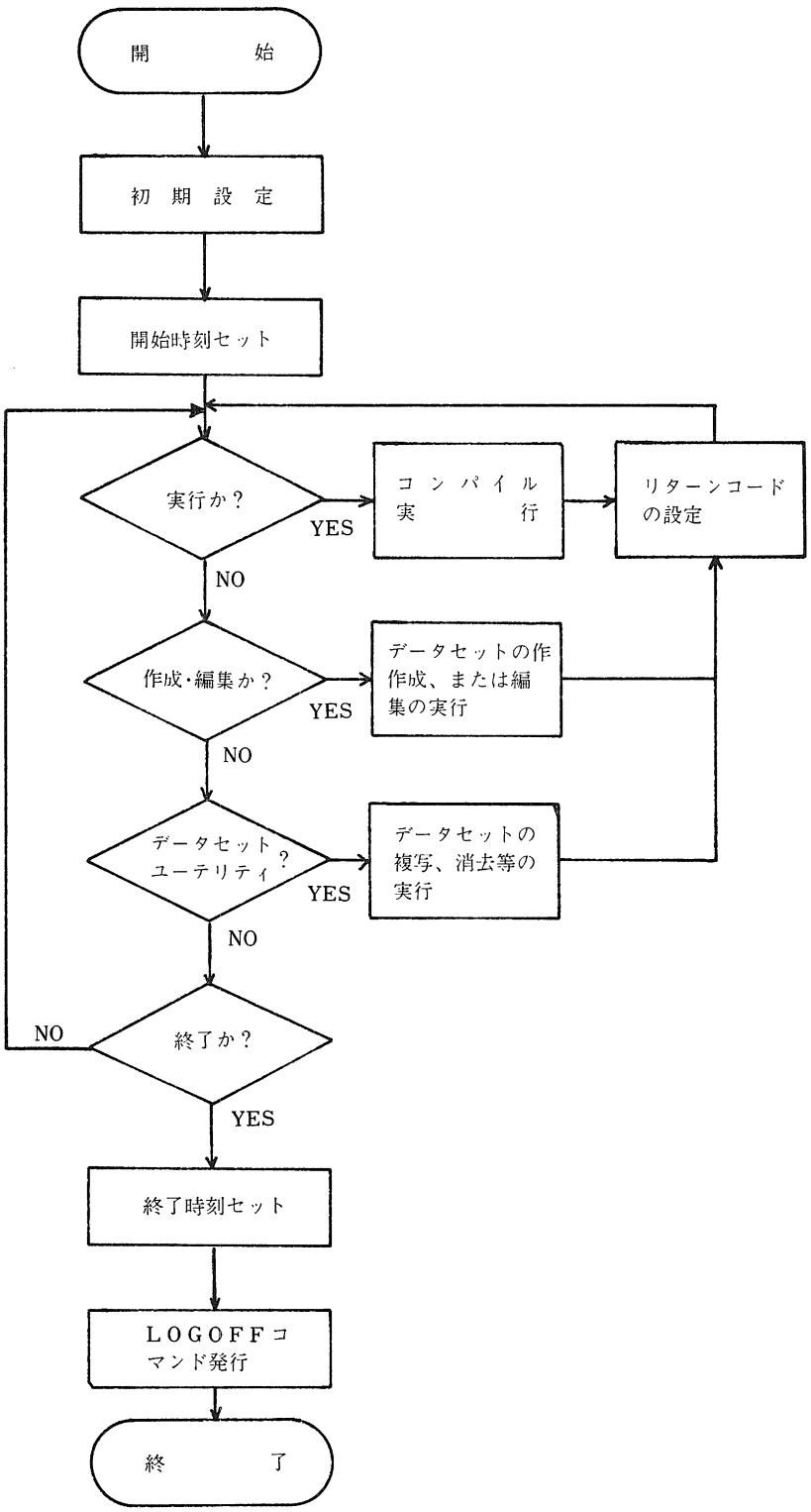

図1　コマンドプロシジャ処理の流れ

同時に多数の学生を実習させるにはかなりの端末を必要とするという設備上の問題，及び TSS 多重度の一時的増加によるシステムの応答時間の悪化が他の研究利用者にも及ぶという問題等がある。しかしこうした問題点は，最初から情報処理教育を目的としている各大学の情報処理教育センターではある程度対策済であり，また将来的に解決可能な問題である。従ってこうした問題点を別にして考えれば，プログラム修正の容易さ，ファイル処理に関する操作，及び資源の消費等の点を考慮すればバッチ処理よりも優位にたっているといえる。特にプログラムの修正，コンパイル，実行といった一連の処理を繰り返し実行することが多い情報処理教育の実習では，TSS 利用による実習がより適していると考えられる。

III　TSS による情報処理実習

TSS による実習を行なうには事前に端末操作，及び TSS コマンド，データセット（ファイル）に関する知識を習得させる必要があることは前述した。このうち TSS コマンドに関しては，コンピュータシステムの OS (Operating System) によって，同じ機能をサポートしていてもシンタックスは微妙に異なることが多い。本学の計算センターのシステムは IBM, HITAC 系のシステムとある程度互換性があるが，それでも TSS のコマンドのシンタックスは一部異っており，NEAC, TOSBAC 系とはさらに異ってくる。従って一部機種のコマンドのシンタックスを覚えさせることに時間をかけるよりも，より簡単な方法で同一の機能が実行できることが，情報処理の専門家を養成する必要のない場合にはより適していると考えられる。今回試作したコマンドプロシジャは，上記のような意味からも簡単な操作性と，利用時の各処理に対するリターンコードの採取といったロギング機能も備えたものである。このコマンドプロシジャ実行時のシステムへの応答は主としてプロンプティングメッセージによる選択入力方式で，通常1文字を選択指定する。実際の実習の流れは TSS 開始直後にコマンドプロシジャ名を入力し，以後このコマンドモード内で処理を続行し，モード終了後即 LOGOFF 状態となる，このコマンドプロシジャの処理の流れ図を図1に示す。図1の流れに従って利用者は会話処理を実行できるが，プログラムテキストの作成，或いは編集時にはシステムの EDIT サブコマンドを使用する。しかし，このサブコマンド以外については前述のとおり簡単な操作で実行できるので，事前にデータセットに関する知識を習得させておけばコマンドに関してはより簡単な説明で TSS の実行が可能と考えられる。

IV　コマンドプロシジャの作成

ここでは本学の計算センターのシステムにおけるコマンドプロシジャの作成について検討する。本学の計算センターのシステムでは TSS は AIF とも呼ばれているが，同じ M シリーズの上位機種の TSS のサブセットと考えることができる。従って一部機能の制限が上位機種に比較すると多いがコマンドプロシジャ作成機能はサポートされている[3]。本学のシステムでコマンドプロシジャを利用するには "CLIST" という識別子を有する区分データセット（または順編成データセット）に，TSS コマンド及びプロシジャ文より構成されるテキストを格納し，"EXEC" コマンドまたはメンバ名をコマンド名として直接指定して実行する。今回は後者の方法を採用する。この方法ではコマンド入力以前にテキストが格納されているデータセットを DD 名 "SYSPROC"

```
LOGON AIF ABCD111                    ←ＴＳＳ　セッションの開始
 :
 :
ALLOC F(SYSPROC) DA(PROC.CLIST)
EAT
 :
 :
```

リスト1　コマンドプロシジャデータセットの利用宣言

```
PROC 0
CONTROL NOFLUSH NOMSG
FREE F(STAT FT06F001 FT05F001)
ATTN DO
GOTO WRITE0
END
ERROR DO
GOTO WRITE0
END
ALLOC F(STAT) DA(STAT.DATA) MOD
ALLOC F(FT06F001) DA(*)
ALLOC F(FT05F001) DA(*)
OPENFILE STAT OUTPUT
SET &STAT=&STR(STAT &SYSUID &SYSDATE &SYSTIME)
PUTFILE STAT
WRITE0:WRITENR RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ?
READ &ANS0
IF &ANS0 EQ R THEN -
DO
   WRITENR DATASET NAME(MEMBER NAME) ?
   READ &DSNAME0
   FORT &DSNAME0 NOGO OBJ(OBJ(MEM))
   SET &STAT=&STR(RUN       FORT &LASTCC)
   PUTFILE STAT
   WRITE COMPILE END RETURN CODE=&LASTCC
   SET &STAT=&STR(RUN           GO   &LASTCC)
   PUTFILE STAT
   GOTO WRITE0
END
IF &ANS0 EQ E THEN -
DO
   WRITENR DATASET NAME (MEMBER NAME) ?
   READ &DSNAME1
   E &DSNAME1
   TERMIN END,END S
   DATA
   END &SYSDVAL
   ENDDATA
   SET &STAT=&STR(EDIT          &LASTCC)
   PUTFILE STAT
```

```
   GOTO WRITE0
END
IF &ANS0 EQ U THEN -
DO
   WRITENR DATASET COPY(C) / DELETE(D) / NAME(MEMBER) LIST(L) / PRINT(P) ?
   READ &ANS2
   IF &ANS2 EQ C THEN -
   DO
    WRITENR FROM ?
    READ &INDS
    WRITENR TO ?
    READ &OUTDS
    COPY &INDS &OUTDS
    SET &CODE=&LASTCC
    IF &CODE EQ 0 THEN WRITE COPY NORMAL END
    SET &STAT=&STR(UTILITY   COPY &CODE)
    PUTFILE STAT
    GOTO WRITE0
   END
   IF &ANS2 EQ D THEN -
   DO
    WRITENR DATASETNAME ?
    READ &DELDS
    WRITENR DELETE &DELDS ? (Y/N) ?
    READ &ANSY
    IF &ANSY EQ Y THEN -
    DO
     DELETE &DELDS
     SET &CODE=&LASTCC
     IF &CODE EQ 0 THEN WRITE DELETE NORMAL END
     SET &STAT=&STR(UTILITY   DEL  &CODE)
     PUTFILE STAT
    END
    GOTO WRITE0
   END
   IF &ANS2 EQ L THEN -
   DO
WRITE1:WRITENR DATASET(D) OR MEMBER(M) ?
   READ &ANSY
   IF &ANSY EQ D THEN -
```

```
   DO
    LISTC
    SET &STAT=&STR(UTILITY   LISTC&LASTCC)
    GOTO PUT
   END
   IF &ANSY EQ M THEN -
   DO
    WRITENR DATASETNAME ?
    READ &DSNAME
    LISTD &DSNAME,M
    SET &STAT=&STR(UTILITY   LISTD&LASTCC)
    GOTO PUT
   END
   GOTO WRITE1
PUT:PUTFILE STAT
    GOTO WRITE0
    END
    IF &ANS2 EQ P THEN -
    DO
     WRITENR DATASETNAME(MEMBER NAME) ?
     READ &DSNAME3
     LIST &DSNAME3
     SET &STAT=&STR(UTILITY  LIST &LASTCC)
     PUTFILE STAT
     GOTO WRITE0
    END
IF &ANS0 EQ OFF THEN -
DO
   ERROR OFF
   DEL OBJ.OBJ
   SET &STAT=&STR(STOP  &SYSUID &SYDATE &SYSTIME)
   PUTFILE STAT
   CLOSFILE STAT
   LOGOFF
END
GOTO WRITE0
EXIT
```

リスト２　コマンドプロシジャの内容

```
EAT
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? E          ←ＥＤＩＴ選択
 DATASET NAME(MEMBER NAME) ? ADM.FORT(TEST)                ←データセット名入力
 E
L
 00010 STOP                                                ←ＥＤＩＴモード
 00020 END
 END OF DATASET
END                                                        ←ＥＤＩＴ終了
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? R          ←プログラムの実行選択
 DATASET NAME(MEMBER NAME) ? ADM.FORT(TEST)
 GE COMPILER ENTERED
 END OF COMPILATION
 COMPILE END RETURN CODE=0                                 ←リターンコード表示
 RUN END RETURN CODE=0                                     ←　　〃
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? U
 DATASET COPY(C) / DELETE(D) / NAME(MEMBER) LIST(L) / PRINT(P) ? C
 FROM ? ADM.FORT(TEST)
 TO ? NEW.FORT(TEST)
 COPY NORMAL END                                           ←コピー終了メッセージ
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? U
 DATASET COPY(C) / DELETE(D) / NAME(MEMBER) LIST(L) / PRINT(P) ? D
 DATASET NAME ? NEW.FORT(TEST)
 DELETE NEW.FORT(TEST) ? (Y/N) ? Y                         ←DELETE確認
 DELETE NORMAL END
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? U
 DATASET COPY(C) / DELETE(D) / NAME(MEMBER) LIST(L) / PRINT(P) ? P
 DATASET NAME(MEMBER NAME) ? ADM.FORT(TEST)
 00010 STOP
 00020 END
 RUN(R),CREATE OR EDIT(E),UTILITY(U),END(OFF) ? OFF        ←終了選択
*** SHIMANE UNIVERSITY COMPUTER CENTER ***                 ←終了メッセージ
                    :
```

リスト３　コマンドプロシジャの実行

で ALLOCATE する必要がある。リスト1にこの例を示す。この方法では利用者は各自の目的に合ったコマンドを作成し,任意の名称(ただしメンバ名の命名規則による。)を付加して実行できる。他のシステムではデータセット名を "USER PROFILE" にセットしておけば,TSS セッションの開始と同時にコマンドプロシジャは有効となる場合がある[4]。リスト2にコマンドプロシジャの内容を示す。

V コマンドプロシジャの実行と統計情報の収集

今回試作したコマンドプロシジャは "EAT" という名称でデータセット "PROC. CLIST" のメンバに作成した。リスト3に "EAT" コマンドの実行例を示す。データセット "PROC. CLIST" の ALLOCATE 後 "EAT" と入力すれば,システムはこれをユーザーコマンドと解釈して実行する。リスト3の実行例において,下線部は端末からの入力を示す。

リスト4は実行終了時の採取された統計情報の内容を示している。採取している情報は以下のとおりである。

1. 利用者課題番号
2. コマンド入力日付,時刻
3. 実行した機能
4. リターンコード
5. 終了時刻

これらの統計情報は事前に用意するデータセットに書きこまれるので,実習開始以前にこのデータセットを作成しておく必要がある。また教育の実習の場合には多数の学生が同時にセッションを開始する可能性があるので,データセットの競合については十分注意する必要がある。

VI むすび

FACOM OS IV/X8 AIF のコマンドプロシジャ機能を利用して情報処理教育用のコマンドプロシジャを試作した。本学の計算センター利用者であれば誰でも作成,利用可能な機能であるので,今後の改良,機能拡張も可能である。ただし,詳細な統計情報が採取できないとか,統計用データセットの競合といった問題が生じる。こうした問題を解決するにはどうしても計算センタールーチンの修正を必要とする場合が多いが,教育,研究,事務といった種々の利用者が存在する計算センターでは,こうした修正は困難である。従って本コマンドプロシジャの存在価値もあるといえる。今回採用したプロンプティングによる方式は,あくまでも初心者用といえるもので,TSS の利用に慣れている立場からみれば直

統計データ ↓	説明 ↓
START ABCD111 08/30/84 14:07:00	←開始 課題番号 年月日 時刻
EDIT 0	←編集 リターンコード
RUN FORT 0	←実行 ステップ名 リターンコード
RUN GO 0	
UTILITY COPY 0	
UTILITY DEL 0	
UTILITY LIST 0	
STOP ABCD111 08/30/84 14:13:03	←終了

リスト4 統計データの内容

接 TSS コマンドを入力したほうが速いと思えるかもしれないが，初期の情報処理教育実習には有効と考えられる。もちろん，プロンプティングのメッセージの表現方法，及び制御の流れ等についてはさらに検討しなければならないが，これについては今後の試用の結果を参考にして検討する予定である。

文　　献

1) 堀口，川添，奈良：〝東北大学 情報処理教育 センター授業援助システム－TESST－〟，信学技報，ET82－5 (1982)
2) 島根大学電算センター広報，No.1 (1982)
3) FACOM OS IV/X8 マニュアル 〝AIF コマンド文法書〟
4) 例えば，HITAC VOS2/3 マニュアル 〝TSS 操作〟