

4. 技術解説

ユーザによる初心者向けTSS利用の手引

教育学部地理学教室

西原 純

私の研究室では、4年生の卒業研究にコンピュータを用いた地理的情報処理を行っており、毎年3～4名が情報処理センターを利用する。ところが、メーカ提供のマニュアルは大変わかりにくく、私自身が多くの時間を費やしてTSSの使用方法を説明しなければならなかった。一度メーカのマニュアルを読んだことのある人なら、このマニュアルが大変わかりにくいことは、誰しも痛感していることだろう。

私が以前に利用していた東北大学大型計算機センターでは、「TSSの使い方」（初版は1978年発行）という180ページの利用の手引があった。この利用の手引によって、初心者でも独習でTSSの使用方法がマスターできた。

残念なことに長崎大学情報処理センターには、独習でTSSの使用方法がマスターできる利用の手引がない。このような手引が刊行されるのを待ち望んでいたが、仲々刊行されそうもないし、学生への使用方法の説明で忙殺される状況は変わらないので、自分でつくることを決心した。利用の手引を作成するほどTSSの利用方法に熟達していないと、私自身は百も承知であったが、私の研究室用の手引として何とか作成することができた。

そして、1986年5月よりこの利用の手引を用いて学生のTSS利用を始めた。同じ教育学部の先生よりこの手引の公開を求められたので、恥を忍んでセンターレポートに掲載して頂くことになった。その際、情報処理センター阪上直美・内本佳彦両氏の協力を頂いてこの手引を再改訂したので、記して感謝の意を表したい。

この利用の手引は初心者を対象としたため、基本的な方法について説明してある。その結果、PFDやコマンドプロシージャの効率的な使用方法については、割愛せざるを得なかった。効率的な使用方法については、近年FACOM大型計算機ユーザのためのTSS利用の手引がいくつか出版されているので、そちらを参考にして頂きたい。

また、長崎大学情報処理センターでも詳しい利用の手引が作成中である。しかし、利用者にとっては、マニュアルは何種類あっても良いものである。特にユーザによるマニュアルは、自分自身のわかりにくかった点や、一連の処理の流れを中心に記述されているので、メーカ提供のものなどと違った視点のマニュアルになるのではなからうか。これを契機に、ユーザーによるマニュアルもたくさん作成されることを願う次第である。

利用の手引の内容については、多くの不十分な点や間違った点があると思われるが、より良いTSS利用の手引にするために、御指摘頂ければ幸いである。

(1986年10月版)

目次

- 1 計算機を利用するために
- 2 セッションの開設・終了方法
 - 2-1 端末の種類とキーボードの操作方法
 - 2-2 セッション開設の手続き
参考1 セッション開設時のリージョンサイズ指定について
 - 2-3 セッション終了の手続き
- 3 データセットの作成
 - 3-1 データセットとは
 - 3-2 プログラムの入力
 - 3-3 データの入力
参考2 区分データセットの取り扱い方法
 - 3-4 会話型リモートバッチ処理のジョブ制御文 (JCL) の作成方法
 - 3-5 データセットの出力 (日本語プリンターの場合)
 - 3-6 データセットの器の作成方法
参考3 データセット作成中にセッションを切断された時の対処方法
- 4 EDIT-FSO によるデータセットの新規作成・修正・追加
 - 4-1 EDIT-FSO によるデータセットの新規作成
 - 4-2 EDIT-FSO によるデータセットの修正・追加
 - 4-3 EDIT-FSO の行サブコマンド
 - 4-4 EDIT サブコマンドによる編集
 - 4-5 PF キーの配列
- 5 プログラムの実行 (EDIT コマンドを用いる場合)
 - 5-1 翻訳 (COMPILE) と実行
 - 5-2 データをキーボードから入力して実行する場合
 - 5-3 データをデータセットから入力して実行する場合
 - 5-4 計算結果のデータセットへの出力 (会話型処理の場合)
 - 5-5 コマンドプロシジャ (マクロ命令) を用いたデータセットからの入出力
- 6 会話型リモートバッチ処理
 - 6-1 会話型リモートバッチ処理とは
 - 6-2 会話型リモートバッチ処理用のジョブ制御文 (JCL)
 - 6-3 バッチジョブのモニター
 - 6-4 SORP 機能 (バッチジョブの取りだし請求)
- 7 データセットの管理
 - 7-1 データセットの編集に関わる EDIT 以外のコマンド
 - 7-2 既存のデータセットの一覧を出力させるコマンド
FLIST コマンド・MLIST コマンドを中心に
 - 7-3 書込み・読みとり許可 (データセットの保護)
- 8 困った時の対処方法

1 計算機を利用するために

長崎大学情報処理センターの大型計算機(FACOM M360)を利用する場合のほとんどは、TSS 端末を用いる会話型処理である。TSS とは、time sharing system の略称であり、大型計算機の動作を多数の端末利用者に順番に割当てる機能のことである。TSS では、非常な短い時間づつ利用者に動作が割当てられているが、超高速で作動しているので、多数の端末利用者にとってもあたかも一人で計算機を占有しているかのような状態になる。

以下に、TSS 利用のための基本的なことから述べてみよう。

センターの計算機は、いつでも誰でも使えるというわけではない。利用者には資格が必要である。この資格を表すものは「課題番号」と呼ばれ、センターの承認を受けた人(教官、大学院生、事務官で利用申請をした者、および情報処理科目の受講生など)に与えられる。課題番号は計算機との会話の開始時や、計算結果の取り出し時に必要であり、他の利用者の JOB と区別するのに最も大切なものである。

次に、「パスワード」が必要である。これは、いわば計算機という金庫を開けるための鍵である。この金庫に物をしまう時、あるいはそこから物を取り出す時に必ず使わなければならない。計算機と利用者間で決められた合言葉であり、利用者が本人かどうか確認するためのものである。従って、パスワードの入力を一度に2回続けて誤ると「利用資格がない者」と見なされ、TSS による計算機利用の要求が拒否されてしまう。課題番号というのは、意外に他人の目にふれやすいが、万一これを他人に使われてもパスワードさえ知られていなければ、まずは安全なので、パスワードはメモしたりせずに確実に記憶しておくことが重要である。

さて、課題番号・パスワードという2つの関所を越えると、いよいよ計算機の利用が可能となる。

2 セッションの開設・終了方法

2-1 端末の種類とキーボードの操作方法

最初にすべきことは、どの端末を使おうとしているかを確認することである。特殊な端末として、日本語端末・英小文字端末・バドミントンプリンター端末などがある。それぞれ少しづつ使用目的と操作方法が違うので注意しなければならない。

ここでは、情報処理センター第一端末室にある通常のディスプレイ装置 (DP) ・日本語ディスプレイ装置 (NDP) を例に述べる。

○ キーボードのモード (シフトロック形式)

(「英数モード」・「英記号モード」・「カナモード」・「カナ記号モード」)

端末の種類を確認したら、次にキーボードを見よう。キーボードのキーの配列は一般のタイプライター・パーソナルコンピュータのものほとんど同じである。ただし、富士通の大型コンピュータのキーボードでは、「英数モード」・「英記号モード」・「カナモード」・「カナ記号モード」の4種類のモードを選択して、1つのキーを4種類のキャラクターに対応させている。この4種類のモードを選択するキーは、キーボード最下段のスペースバーの両端にあり、いずれも一度押すとそのモードにロックされているので注意すること (キーのモードランプが点灯する、ただし端末の電源をいれた後)。そのモードを解除して他のモードに移る場合には、他のモードキーを押せばよい。

TSS 処理のコマンドなどでは、ほとんど「英数字モード」で用が足りる。ただし (,) , = などは「英字記号モード」を使用しなければならない。「英数字モード」と「英字記号モード」との間のモードの変更は、その都度希望するモードキーを押さなければならないので、いささかやっかいである。

○ DELT キーと INSTMODE キー

キーでさらに注意が必要なのは、DELT キーと INSTMODE キーである。DELT キーを押すとカーソルのある位置の文字が削除される。そして、DELT キーを押し続けると次々にカーソルよりも右にある文字が連続的に削除されてしまう。INSTMODE キーを押すとディスプレイ画面の最下段に「^」が表示される (ただし端末の電源をいれた後)。

RA

ALPHA ^ . 2

この状態を INSERT モードという。INSERT モードでは、新たに文字キーを押すたびにカーソルのある位置の前に次々文字が挿入される。この INSERT モードを解除するには、キーボード左下の RESET キーを押す。

2-2 セッション開設の手続き

TSS の起動から終了までをセッションと呼ぶが、セッションを開設して TSS が使用できる状態にしよう。

まず、端末の右部にある電源を ON にする。使用可能ランプ (黄緑) の点灯を待って ENTER キー (以下、[←] とする) を押す。メッセージ表示 (「SYSTEM READY」) が出たら、キーボードを操作して LOGON コマンドと課題番号を入力して、[←] を押す。計算機はパスワードの入力を促すので、指示に従って入力 (画面には表示されない) し、[←] を押す。以下、セッションが開設され、センターからのニュースなどが表示されて READY という状態になる。この状態で各種のコマンドを入力することによって、コンピュータを操るわけである。

また、使用しようとした端末の電源が入っている場合には、他人が使用中であるかもしれない (計算結果を取りに行っているなど)。但し、画面に「SYSTEM READY」が表示されていれば、使用可能である。

最初のうちは、キーボードの操作になれていないので落着いてゆっくりキーを押すこと。また、途中で押し間違った場合には、画面最下段のエラー表示に注意し 8 「困った時の対処法」を参照すること。

- 手順 (下線上の文字は、計算機側からの表示)
(1) 電源ON (端末ディスプレイ右側下にある)
即座に黄色の電源ランプが点燈する。

- 次に
(2) 黄緑色の使用可能ランプが点燈する (数秒間かかる)

日本語端末室のワープロ端末装置 (WDS) では、電源ONの後数十秒間かかって画面最下段のモード表示が下のようになり英数字モードになって、はじめて (2) の使用可能の状態になるので、注意が必要である。

RA

ALPHA 2 英数字

- (3) [←] (何も入力しないで [←] を押すことを空 ENTER という)
接続要求に対する準備が完了したという表示が (4) に出るので、

- (4) JCET005 SYSTEM READY

次に課題番号をキーボードから (F1234を課題番号とした場合)

LOGON TSS F1234

とタイプし、 [←] を行なう。

- (5) +PASSWORD ?

と表示されるので、PASSWORD をタイプして [←] を行なう。PASSWORD 入力時のみタイプした文字は画面には表示されずカーソルのみ移動する。

入力ミスの場合、再入力の指示が出る。また、 [←] 前ならば、カーソル移動キーで戻っても訂正可能。PASSWORD が正常に受け付けられると、計算機からセッションが開設された旨のメッセージが下記のように表示される。

なお、長崎大学情報処理センターでは一つのセッションは2時間以内に制限してある。

- (6) KDS70001I F1234 LAST ACCESS AT
KEQ56455I F1234 LOGON IN PROGRESS AT
JOB NO = TSU1234 CN(01)
KEQ56451I NO BROADCAST MESSAGES
READY (コマンドモード)

ここまでの処理によって、端末は READY が表示されてコマンドモードと呼ばれる状態になった。以下、実際のプログラム・データの入力・修正・実行を行なうためのいろいろなコマンドの入力が可能である。まず簡単なプログラムを入力し、実行させて見よう (詳細については、後の項目を見ること)。このような利用形態を会話型処理という。他に TSS のもとで一括処理するバッチ処理 (会話型リモートバッチ処理) があるが、この方法については後で詳しく述べる。

以下、例にしたがって英文字・数字・カナ文字・記号などを端末のキーボードよりタイプするわけであるが、上述のようにキーボードの操作方法にはくれぐれも注意が必要である。

平均値を求める FORTRAN 77 のプログラム (14 行ある) を MEAN.FORT77 という名前のデータセットに作成し、データの個数とデータそのものをキーボードより入力して実行させてみよう。この方法は最も初歩的な使用方法で、パーソナルコンピュータの使い方に似ている。

- 手順
(1) READY (コマンドモード)
EDIT MEAN FORT77(FIXED) NEW [←]

EDIT はプログラム・データを入力・修正・実行・保存するためのコマンドで、TSS 処理のうち最も基本的なコマンドである。そして、コマンドというものは単独で用いられるものもあるが、ほとんどはオペランドと呼ばれる一種のキーワードとともに用いられる。この場合 (

EDIT コマンド)では、FORT77(FIXED) と NEW がオペランドで、これから編集するデータセットが FORTRAN77 の固定形式の新しいデータセットであることを意味している。

(2) EDIT
INPUT (サブコマンドモードで入力待ち状態である)
00010 _

「_」のカーソル位置の後に、第 1 行を入力する。

```

00010 *      CALCULATION OF MEAN      [←]
00020      DIMENSION SCORE(100)      [←]
00030      READ(5,*) NUMBER          [←]
00040      DO 100 I=1, NUMBER         [←]
00050          READ(5,*) SCORE(I)    [←]
00060      100 CONTINUE               [←]
00070          SUM=0.0                [←]
00080      DO 200 I=1, NUMBER         [←]
00090          SUM=SUM+SCORE(I)      [←]
00100      200 CONTINUE               [←]
00110          AMEAN=SUM/NUMBER      [←]
00120          WRITE(6,*) AMEAN      [←]
00130          STOP                  [←]
00140          END                    [←]
00150      [←]

```

第 14 行が最終行なので、15 行めは [←] のみ。この措置によって入力待ち状態が終了する。次にプログラムを実行させるサブコマンド RUN をタイプする。また、プログラムの入力時にミスタイプした時には、そのまま入力を受け入れ完了後に後述するプログラムの修正方法を用いるとよい。

(3) EDIT (EDIT コマンドモード)
RUN [←]
FORTAN 77 COMPILER ENTERED
END OF COMPILATION

行番号 30 の READ 文にもとづく入力促進記号「 00030 ? _ 」が画面に表示されるので、以下データの個数を入力する。

```

00030 ?  

_ 5 (データの個数) [←]

```

するとさらに、行番号 50 の READ 文にもとづく入力促進記号「 00050 ? _ 」が画面に表示されるので、以下データそのものを 1 行に 1 個ずつ順次入力する。

```

00050 ?  

_ 1.23 [←]  

00050 ?  

_ 4.74 [←]  

00050 ?  

_ 5.69 [←]  

00050 ?  

_ 2.84 [←]  

00050 ?  

_ 0.81 [←]  

3.06199932  

END OF GO, SEVERITY CODE = 00  

EDIT  

SAVE [←]

```

(4) KEQ52460I SAVED IN DATA SET 'F1234.MEAN.FORT77'
EDIT (サブコマンドモード)

END [←] (EDIT 状態を終了させるサブコマンド)
(5) READY (コマンドモード)

以下、必要に応じてプログラム修正のための EDIT コマンドや、保存したプログラムの入ったデータセット名を確認する LISTC コマンドなどをタイプする。
TSS 処理を行なう際には、

- (1) 「コマンドモード」= (画面には READY と表示されている状態)
コマンド (例えば、EDIT コマンド) が受付られる
- (2) 「サブコマンドモード」= (画面には EDIT などのコマンド名が表示されている状態)
サブコマンド (例えば、EDIT 状態の FS サブコマンド) が受付られる
ただし、サブコマンドモードでも
「X コマンド」という型で、コマンドも入力可能である
- (3) 「データモード」= (画面には 行番号 ? と表示されている状態)
データを入力できる状態

の三つのモードがある。現在のモードと受付可能な命令を常に頭に入れておく必要がある。
またこれらの三つのモードからそれぞれ前のモードにもどすには、PF3 キーや END サブコマンドを使用する。さらに、緊急の場合には [ALT] キーと [DUP] キーを同時に押すアテンション割り込み機能を使用する。

READY (コマンドレベル)
必要なコマンドの入力、[←]
コマンドモード表示 (サブコマンドレベル)
..... 以下、サブコマンド入力可能
サブコマンド入力 [←]
:
:
:

以上のような手順によって計算機の利用ができる。但し、1セッションの開設時間は2時間と限られているので、大変面倒ではあるがセッションは必ず2時間以内に LOGOFF コマンドを用いて終了させなければならない。以下に、LOGOFF の手順を述べる。もし、時間内に終了させなかった場合には、利用者にとっては突然に、また強制的にセッションを断ち切られる。それがデータセットの作成中だった場合の対処法については、後述の参考3で説明する。

-----参考1-----

・セッション開設時のリージョンサイズ指定について

TSS 利用の場合、リージョンサイズは標準値 256KB が自動的に割り当てられている。初歩的計算のうちには意識する必要はない。ただし、大規模な配列を有するプログラムを実行する時には、標準値のリージョンサイズでは overflow する恐れがあるので、予め大きなサイズを指定しておくといよい。その際、4104KB (4MB)まで指定が可能である。

例えば、統計パッケージ ANALYST を使用する場合には 3MB (1024KB*3) 以上、自動翻訳プログラム ATLAS I・ATLAS II を使用する場合には 4MB (1024KB*4)、日本語処理を行なう場合には 768KB 以上のリージョンサイズが必要である。

なお、リージョンサイズ指定はセッション開設時しか出来ない。

リージョンサイズを 1024KB (1MB)にとる場合

LOGON TSS F1234 S(1024) [←]
(サイズ指定は KB 単位)

2-3 セッション終了の手続

コマンドモードでの操作を既に完了し、セッションを終了したい時には、READY 状態に戻す (ex. EDIT コマンドモードで END サブコマンド [←] とする等)。READY 状態で LOGOFF コ

マンド [←] を入力する。これが受け付けられると、使用時間・日時 (及び料金) 等の表示がなされる。ここで、そのセッションは切断されたことになる。更に [←] を押すと、セッション開設前のメッセージ表示「SYSTEM READY」が出て、次のセッションを待つ状態になる。そこで、電源を切る。(継続使用の場合には電源を切る必要はなく、続けてセッション開設の手続を行えばよい。)

端末の操作方法を間違った場合に、端末がある画面に固定された状態になってしまい、種々のコマンドや LOGOFF コマンドを入力しても受付られず、手に負えない状態になってしまう場合があるが、絶対に端末モニターのスイッチを切断してはならない。切断すると端末に何も映らず、セッションが終了したように見えるが、実は計算機内ではセッションは開設されたままの状態に陥る場合がある。このような状態では、計算機本体に大きな障害を与える恐れがある。端末の操作方法を間違った場合には、エラー処理を何種類か試み (第8章)、それでも回復できない場合にはセンターの職員に連絡すること。

下に、課金情報も出力させる LOGOFFC コマンドの使用例を示す。

手順

```
(1) READY
LOGOFFC [←]
(2) RETURN CODE : 0000
CPU TIME( 5.08SEC.) USE TIME( 12MIN.) REGION SIZE( 312KB)
INPUT( 40LINES) OUTPUT( 84LINES) EXCP( 161TIMES)
SESSION CHARGE SINCE(TSU1234, 13:16:31) 34YEN
TOTAL CHARGE SINCE 04/01/86 (EXCEPT THIS SESSION'S) 1,135YEN
KEQ56470I F1234.LOGGED OFF AT 13:28:27 ON APRIL 16,1986+
KEQ54220I SESSION ENDED
***
```

画面に *** が表示された状態で、セッションは終了した。しかし、再度セッションを開設する場合も多いので、ENTER キーを押しておく。

なお利用中に、画面に *** がしばしば表示される。この状態では、そのまま待っていても画面は変化しないし、コマンドをタイプしても受け付けられない。これを解消するためには、ENTER キーを押す必要がある。

```
[←]
(3) JCET005 SYSTEM READY
..... セッション開設前に戻る
(4) 電源OFF
```

(1) の「LOGOFFC」コマンドは、セッション終了時にこれまでの課金情報を表示させるコマンドである。また、継続使用の場合には、(4) は不要である。引続き、LOGONコマンドでセッションを再開すれば良い。

TSS の使用料金は意外に高いので、残りの予算額に注意すること。また、プログラムの修正などで端末をいじらない場合には、LOGOFF したほうが良い。セッション開設時間1分間につき1円加算される。

3 データセットの作成

3-1 データセットとは

データセットとは、計算機の外部記憶装置（磁気ディスク装置）に保存してあるデータやプログラムなどから構成された記録の集合体である。すなわちデータセットは、これらのデータやプログラムの塊を、個別の名前を付けて保存している書物のようなものである。1つのプログラムを実行する時に、必要なプログラムやデータをいちいちその場でキーボードから入力することは、プログラムが長かったりデータ数が多くなると大変面倒なことになるであろう。そのような時のために、予じめプログラムやデータのデータセットを作成しておくことと便利である。

○ データセットの形式

データセットには、

- (1) 順データセット (略称 PS)
- (2) 区分データセット (略称 PO)

の二つの形式がある。順データセットは、データセットの中のデータが処理される順番に格納されているもので、一つのプログラムやデータが格納できる。区分データセットは、いくつかのデータセットが一つの代表名（区分データセット名）のもとに、格納されているものである。それぞれのデータセットはメンバー名で区別される。同じ種類のデータを格納する際には、区分データセットの方が整理上都合が良い。

データセットの作成には、READY が表示されているコマンドモードで EDIT コマンドを用いる。ここでは、2種類のデータセットのうちで操作の簡単な順データセットを作成する場合を取り上げ、プログラムの作成の手順について述べてみる。

○ データセットの名前

データセットには個別に名前を付ける。この名前によって他のデータセットと区別することができる

順データセットの場合には、その正式な名前は

- (1) 課題番号
- (2) データセット名 (英文字を先頭とする8文字以内の英文字・数字で構成)
- (3) 内容識別修飾子 (データセットの内容を示す FORT77、DATA、CNTL、CLIST など)

の3つの部分からなり、

'F1234.AREA.FORT77'

という表現形式をとる。

ただし、開設されたセッションと同一課題のデータセットを指定する場合には、

- (2) データセット名
- (3) 内容識別修飾子

の2つで充分であり、

AREA.FORT77

と表現する。

また、データセットの正式名称を用いる場合には必ず

'課題番号.データセット名.内容識別修飾子'

ように「」(SINGLE QUOTATION MARK、キーボードでは 英記号モードキーを押した後、7キーを押す)を両端に付ける。他の課題番号のデータセットをコピーする場合に用いる。

区分データセットの正式な名前は、

- (1) 課題番号
- (2) データセット名
- (3) 内容識別修飾子
- (4) メンバー名 (英文字を先頭とする8文字以内の英文字・数字で構成)

の4つの部分からなり、

'F1234.PROGRAM.FORT77(MEAN)'

という表現形式をとる。

○ データセットの新規作成

データセットの作成には、READY 状態で EDIT コマンドを用いる。ここでは、2種類のデータセットのうちで操作の簡単な順データセットを作成する場合を取り上げ、プログラムの作成の手順について述べてみる。

ただし、FORTRAN プログラムやデータの多くの場合には入力すべきカラム位置が規定されていたり、指定されてしまう。その際にはデータセットの新規作成でも、入力するカラム位置が

非常にわかりやすいフル・スクリーン形式（EDIT-FSO エディット-フルスクリーンオプションという）の編集機能を用いた方がよい。

EDIT-FSO による新規作成・修正・追加の方法については、4 章で述べる。

3-2 プログラムの入力

EDITコマンドと共に、データセット名及び内容識別修飾子とプログラムの形式指定を入力して [←] すると、新規作成か既成のものかを確認するメッセージが表示される。新規の場合には、INPUT という表示と共に、00010 という第 1 行を意味する行番号が表示される。ここで、用意してあるプログラムを入力し始める。

第 1 行の最後で [←] すると、00020 が表示されて第 2 行へ移る。この操作を必要な行数の入力が終るまで繰り返す。最終行を入力し終っても行番号が表示されて入力を促すので、[←] を行なう。

今度は EDIT という表示になる。そこで、プログラムを保存するために SAVE [←] とし、さらに END [←] とする。これで、プログラムは保存される。

ここでは、プログラム言語を FORTRAN 77、形式を FIXED とする（この他の形式には FREE がある）。形式を「FIXED」とすると固定長レコード（カードイメージ）のデータセットが、「FREE」とすると可変長レコードのデータセットが作成される。一般には、固定長レコード形式を用いる。

プログラムの場合、入力できるのは各行とも 72 カラムまでである。73 ~ 80 カラムの位置には、自動的に行番号が入ることになっている。ただし、LIST コマンドで端末や日本語プリンタにプログラムを出力させると、行番号は自動的に最も左側へ移されて表示される。

7 行ある FORTRAN 77 のプログラムを AREA.FORT77 という名前のデータセット（固定長レコード形式）に作成してみよう。

手順

(1) READY
EDIT AREA FORT77(FIXED) NEW [←]

(FIXED) オペランドを省略すると自動的に (FREE) オペランドを指定したことになり、データセットのレコード形式が異なり、そのままではプログラム中の継続行記号が文法エラーになるので注意が必要である。

(2) EDIT
(3) INPUT (INPUT モード)
00010 _

「_」位置以後に、第 1 行を入力する。

<u>00010</u>	*	CALCULATION OF AREA	[←]
<u>00020</u>		INTEGER A,B,S	[←]
<u>00030</u>		READ(5,*) A,B	[←]
<u>00040</u>		S=A*B	[←]
<u>00050</u>		WRITE(6,*) A,B,S	[←]
<u>00060</u>		STOP	[←]
<u>00070</u>		END	[←]

第 7 行を最終行とする。第 8 行には空行のまま「ENTER キー」を押すと入力モードが終了する。

00080 _ [←]
EDIT (サブコマンドモード)

ここで入力モードが終了して、EDIT コマンドモードにかえた。ところが間違っ、再び空行のまま「ENTER キー」を押してしまった。これはよくある間違いである。

EDIT (EDIT コマンドモード)
[←] (間違っ「ENTER キー」を押したところ)

EDIT
INPUT (INPUT モード)
00080

当然のことながら再び入力モードになる。この場合にはどうしたら良いか考えなさい。そして考えた結果ある処理をすると無事 サブコマンドモードに帰る事ができる。(正解 もう一度 ENTER キーを押す)

- (4) EDIT (サブコマンドモード)
SAVE [←] (内容を保存するためのサブコマンド)
- (5) KEQ52460I SAVED IN DATA SET 'F1234.AREA.FORT77'
EDIT (サブコマンドモード)
END [←] (サブコマンドモードを終了するサブコマンド)

(5) で SAVEサブコマンドを入力せずに END [←] とした場合には、保存しなくても良いのか確認してくる。改めて、SAVE [←] と入力するか NOSAVE [←] とした上で、END [←] しなければならない。従って、入力したデータセットを保存したくない時には、NOSAVE というオペランドを END サブコマンドとともに入力する。

この他、作成したデータセットの名前を変更して保存するには、

EDIT
SAVE 別の新しいデータセット名.FORT77 [←]

と入力すれば良い。

データセットを保存した時には、確実に保存されたかを確認する必要がある。

- (6) READY (コマンドモード)
LIST AREA.FORT77 [←] (データセットの内容を表示させるコマンド)

と入力すれば、保存されているはずの内容が画面に表示される。

3-3 データの入力

大量のデータを扱う処理を行ないたい場合には、データ用のデータセットを作成した上で処理プログラムを実行する時に、そのデータを読み取らせれば便利である。その作成方法は、プログラムの作成とはほぼ同様の手順である。ただし、EDITコマンドの入力時に、内容識別修飾子の後にオペランド NONUM を入力することによって、本来なら行番号が格納される領域である 73 ~ 80 カラムにもデータを入力できるようになる。

ここで、NAGASAKI.DATA というデータセットを作成してみよう。

手順

- (1) READY
EDIT NAGASAKI DATA NEW NONUM [←]
- (2) INPUT
- 3.14 6.98 2.69 [←]

第一行の内容を入力した後、順次各行のデータを入力する。

```
2.95 9.37 [←]
5.24 8.63 [←]
8.71 2.57 [←]
4.33 6.91 [←] .....最終行の内容を入力
[←] .....空行のまま
```

- (3) EDIT
END SAVE [←] (データセットの保存)
- (4) KEQ52460I SAVED IN DATA SET 'F1234.NAGASAKI.DATA'
READY

保存しなくても良い場合には、(3) の EDIT コマンドモード時に END NOSAVE (END NO

でも良い) と入力する。

-----参考2-----

・区分データセットの取り扱い方法

先に述べたように、区分データセットの正式名称は、「課題番号. データセット名. 内容識別修飾子 (メンバー名)」である。

EDIT コマンド時に区分データセットを指定する方法は、区分データセットの一般の名称形式とかなり異なるので、充分注意すること。

プログラムのデータセット PROGRAM.FORT77(MEAN) を新規作成する場合には、メンバー名を指定する位置が一般に名称を指定する場合と異なり、

```
READY
EDIT PROGRAM(MEAN) FORT77(FIXED) NEW [←]
```

というように、メンバー名をデータセット名に続けて () を用いて書く。

データのデータセット JAPAN.DATA(CITY) を指定する場合には、

```
READY
EDIT JAPAN(CITY) DATA NONUM [←]
```

となる。

このように、区分データセットの取扱いは複雑で初心者には難しい。しかし、MLIST コマンド (7-2 「既存のデータセットの一覧を出力させるコマンド」) を用いるとデータセットの編集や実行において著しく使い易く、またデータセットの整理がしやすいので区分データセットを積極的に使用するとよい。

ただし、区分データセットの複写 (COPY コマンド) ・削除 (DELETE コマンド) を行なう場合には、順データセットの場合と異なるのでくれぐれも注意が必要である。

3-4 会話型リモートバッチ処理のジョブ制御文 (JCL) の作成方法

会話型リモートバッチ処理とは、必要な手順を一切計算機にまかせて行なう一括処理のことであり、一つの処理のために会話型処理のように次々正しい順序にコマンドを投入する方法に比較して、手数がかからない便利な処理形式である。必要な手順は JCL という一連の命令によって指示する。JCL は、内容識別修飾子として「CNTL」を持つ独自のデータセットに作成する。

では、プログラムのデータセット (MEAN.FORT77) とデータのデータセット (POP.DATA) を使用するバッチ処理のための JCL を BATCH.CNTL という名前で作成してみよう。なお、この JCL は行番号のないデータセットの方が都合の良い場合があるので (統計パッケージ BMDP を用いる場合など)、EDIT コマンドの入力時に「NONUM」オペランドを添えて、JCL を作成する。

手順

```
(1) READY
    EDIT BATCH CNTL NONUM NEW [←]
(2) EDIT
    INPUT
    //F1234A JOB CLASS=C,REGION=1024K [←]
    // EXEC FORT77,STEP=CLG [←]
    //FORT.SYSIN DD DSN=F1234.MEAN.FORT77,DISP=SHR [←]
    //GO.FT05F001 DD DSN=F1234.POP.DATA,DISP=SHR [←]
    // [←]
    - [←] (ENTER キーのみ)
```

```
(3) EDIT
```

END SAVE [←]
(4) KEQ524601 SAVED IN DATA SET 'F1234.BATCH.CNTL'
READY

ただし、会話型リモートバッチ処理に用いる JCL はよく使うので、色々な処理のために複数のメンバー名を持つ CARD.CNTL という区分データセットに整理して作成しておくのが効率的である。6章「会話型リモートバッチ処理」ではこの方法で説明してある。また、後述するコマンドプロシジャのデータセットの作成方法（5-5「コマンドプロシジャ（マクロ命令）を用いたデータセットからの入出力」）は、リモートバッチ処理 JCL のデータセットとほぼ同様で、内容識別修飾子 CLIST を用いる点異なるだけである。

3-5 データセットの出力（日本語プリンターの場合）

データセットの内容をひととおり入力してしまったら、早速日本語プリンター（NLP）へその内容を出力してみる。NLP は、1階のオープン入出力室と2階の入出力機器室にある。データセットの内容を点検するためには、紙上でチェックするほうがよい。これは、モニター画面でのチェックが目を疲れさせることや、セッションの接続料金がかさむことなどからである。この時に用いるのは、READY 状態での LIST コマンドである。

また、大量なデータセットを出力させる場合や濃縮印刷で出力させる場合には、そのためのバッチ処理があるので、JCL を予め作成しておくことと良い（CARD.CNTL(PRINTPS)・CARD.CNTL(OMPPS)など）。

LISTコマンドの使い方

○ 基本的な使用方法

READY
LIST MEAN.FORT77 SY(D) [←]
または、SY(K)

と入力すると、端末の画面に

KEQ52801I DATA SET F1234.MEAN.FORT77 BEING PROCESSED

と表示され、即座に NLP に出力される。

オペランド SY(D) は、1階オープン入出力室の NLP を出力先に指定している。これを指定しないと、端末の画面上にデータセットの内容が出力される。また、2階入出力機器室の NLP を指定する場合には、SY(K) とする。

○ 行番号なしのデータセットの場合

行番号なしのデータセット（たとえばデータのデータセット）場合、NONUM オペランドを指定しないと、73-80カラムの数字が行番号とみなされて、左端に移されて出力されてしまっていて、表示されたデータセットの形式が思わぬ順序になっているので注意すること。

READY
LIST データセット名.内容識別修飾子 SY(D) NONUM [←]

○ 大容量のデータセットの場合

大容量のデータセットを出力させる場合、TSS の出力ページ数・行数には制限があるので、出力させるデータセットを部分的に指定するほうがよい。

READY
LIST データセット名.内容識別修飾子 N1 N2 SY(D) NONUM [←]

N1 : 出力させたい最初のレコード番号
N2 : " 最後のレコード番号

レコード番号は、行番号なしのデータセットに計算機側が便宜的に EDIT-FSO 画面上で表示してくれる番号とは無関係で、先頭からのレコード数である。
データセット作成の基本的な手順は以上であるが、くれぐれも注意しなければならないのは、

セッションの開設時間である。せっかくのデータセットなのに、作成中にセッションを打ち切られると面倒なことになる。従って、できる限り効率的に内容を SAVE [←] し、2 時間以内毎に LOGOFFC することが重要である。不注意やアクシデントで万が一セッションを打ち切られても、データセット作成中だった場合には対処法があるが(参考3)、プログラムの実行中の場合には実行はやり直しとなる。SAVE と LOGOFFC はこまめに行うべし。

3-6 データセットの器の作成方法

データセットは、EDIT コマンドを用いて作成することが多く、器のみ作成する場合は非常に少ない。ただし、後述するように、センター 2F にある OCR 端末を利用したフロッピーディスクからホストコンピュータへのデータセットの送信を行なう場合や、プログラムの実行結果をデータセットに書き込んで保存する場合などでは、予めデータセットの器だけを作成しておかなければならない。

○ EDIT コマンドを用いる簡便法

データセットの器を作成する最も簡単な方法は、EDIT コマンドを用いて空白行一行を入力し、すぐに END SAVE と入力する方法である。

- (1) READY
EDIT UTUWA DATA NONUM NEW [←]
- (2) INPUT
- [←]

すぐに空白行のまま、ENTER キーを押したので、INPUT モードが終了し EDIT 状態となり、

- (3) EDIT
END SAVE [←]
- (4) KEQ52553I SAVED, BUT DATA SET IS EMPTY
READY

UTUWA.DATA というデータセット名を持つ空のデータセットが作成された。

○ ATTRIB , ALLOC コマンドを用いる方法

上記の EDIT コマンドを用いる方法は非常に簡単であるが、作成されたデータセットの属性は予めシステムによって規定されてしまっている。そのため、既定量の SPACE 量や異なった RECORD 長・BLOCK 長のデータセットを作成するには、この方法が必要となる。では、もう一度 UTUWA.DATA というデータセットを作成しよう。

手順

- (1) ATTRIB コマンド
READY
ATTRIB PS1 LRECL(80) BLKSIZE(3120) RECFM(F B) DSORG(PS) [←]

まず、これから作成しようとするデータセットの属性を代表名 PS1 という名前に ATTRIB コマンド用いて定義したところである。

オペランドの説明

LRECL : レコード長。通常はカードイメージの 80 バイトを指定。

BLKSIZE : レコードの集合体のことをブロックと呼ぶ。データセットの読み書きはブロック単位で行なわれる。しかし、我々はあまりブロック長を意識する必要はない。ここでは、通常の 39 レコード = 1 ブロックを指定する。

ブロック長の情報は、MT (磁気テープ) を使用する場合に必要である。

RECFM : データセットの形式を指定する。通常は、固定長ブロック化形式の F B を指定する。

ANALYST の BANK を作成する場合などには、可変長非ブロック化形式の V S を指定する。

DSORG : データセットが
順データセット (PS)

区分データセット (PO)
のどちらかを指定する。

この ATTRIB コマンドによってなされた定義は FREE コマンドが投入されない限り、セッション終了まで有効である。次に、この PSI というデータセット名の定義を用いて、ALLOC コマンドによってデータセットの器を作成する。

(2) ALLOCATE コマンド

ALLOC コマンドとオペランドが画面 1 行に入り切らない場合には、行末に - (マイナス、または + プラス でも良い) を補い、次の行に、続けて残りのオペランドを入力する。

READY

```
ALLOC DA(UTUWA.DATA) NEW TRACKS SPACE(10,10) RELEASE - [←]  
CATA USING(PSI) [←]
```

オペランドの説明

DA: これから作成しようとするデータセット名。内容識別修飾子まで含む形式で指定する。

NEW: このデータセットが新規作成であることを示す。

TRACKS: 次に指定する SPACE をトラック単位で指定することを意味する。
1トラックは約 300 行 (1行 80 カラム) に相当する。

SPACE: データセットの大きさを
SPACE (初期量、増加量)

で指定する。データセット作成後、後述する方法で出力結果をデータセットに書き込む過程で、データセットのスペースが初期量だけでは不足する場合、与えられた増加量に従って拡張される。拡張回数は 15 回までしか行われないので、大きな出力量の場合には、増加量を大きく取るなどの注意が必要である。

RELEASE: 拡張された残りのスペースを解放する。

CATA: 作成されたデータセットをカタログして保存することを指定する。

○ コマンドプロシジャ (マクロ命令) によるデータセットの器の作成方法

ATTRIB, ALLOC コマンドを使用してデータセットを作成する方法は、非常に繁雑なので、

```
FREE ALL  
ATTRIB PSI LRECL(80) BLKSIZE(3120) RECFM(F B) DSORG(PS)  
ALLOC DA(UTUWA.DATA) NEW TRACKS SPACE(10,10) RELEASE CATA USING(PSI)  
FREE ALL
```

という処理手順を一括したコマンドプロシジャのデータセットを、COMMAND.CLIST(FILECR) という名前を使用してデータセットに予め作成しておくたいへん便利である。コマンドプロシジャの区分データセットの利用方法は、5-5 「コマンドプロシジャ (マクロ命令) を用いたデータセットからの入出力」で詳しく解説してあるのでそちらを参照すること。

----- 参考 3 -----

・データセット作成中にセッションが切断された場合の対処方法

まず、画面を見よ。終了時の表示が出ているはずである。そこで、セッションを再開する。READY 状態で LISTC コマンドを使い、データセット一覧表を見る。その中に、EDITSAVE.内容識別修飾子 という見慣れないデータセット名がきっと有るはずである。READY 状態なので、EDIT コマンドで EDITSAVE.内容識別修飾子 というデータセットを呼び出す。さらに、EDIT-FSO 画面でその中身を確認して [PF3] キーを押す。END サブコマンドで READY 状態へ戻す。ここで確認したデータセットの内容は、恐らく時間切れまで作成中であつたデータセットの内容と同じはずである。

次に、切断時に作成中だったデータセットは不要となるので DELETE コマンドを使って消去する。そして、RENAME コマンドを用いて EDITSAVE.内容識別修飾子 を本来作成中であつたデータセット名に付けかえる。この後、EDIT-FSO 画面できちんと本来のデータセット名で復旧しているかを確認する。

以上が、データセットの回復方法である。以後は、再び自由に入力ができる。

手順

- (1) セッション終了時の表示
[←]
- (2) JCET005 SYSTEM READY
LOGON TSS F1234 [←]
PASSWORD
パスワードの入力 [←]
セッション開設時のメッセージ
- (3) READY
LISTC [←]
- (4) INCATALOG:SYS1.EDF.CMAT
.....
F1234.EDITSAVE.DATA (データの場合)
.....
F1234.EDITSAVE.FORT77 (プログラムの場合)
.....
- (5) READY
EDIT EDITSAVE 内容識別修飾子 [←]
- (6) EDIT
FS [←]
- (7) EDIT-FSO 画面 (内容の確認)
- (8) EDIT
END [←]
- (9) READY
DELETE 切断時に作成中のデータセット名.内容識別修飾子 [←]
- (10) READY
RENAME EDITSAVE.内容識別修飾子
切断時に作成中のデータセット名.内容識別修飾子 [←]
- (11) READY
EDIT 切断時に作成中の 内容識別修飾子 [←]
- (12) EDIT
FS [←]
- (13) EDIT-FSO 画面 (内容の確認)

この処理で、切断時に作成中だったデータセットの作業を再開することができる。

4 EDIT-FSO によるデータセットの新規作成・修正・追加

4-1 EDIT-FSO によるデータセットの新規作成

プログラムやデータを保存するためのデータセットの新規作成方法については、前章ですでに述べた。しかし、FIXED 形式の FORTRAN77 プログラムや大量のデータのデータセットを新規作成する場合にも、常に入力しようとする位置がはっきりわかるフルスクリーン形式の入力形式の方がやさしい。

FACOM - M360 システムではフルスクリーン形式の編集援助システムとして EDIT-FSO が用意されている。EDIT-FSO には豊富な機能を持つ行サブコマンド・EDIT サブコマンドも用いることができる。詳しくは 4-3 「EDIT-FSO の行サブコマンド」と 4-4 「EDIT サブコマンドによる編集」で述べるので、そちらを参照すること。

EDIT-FSO を用いる場合には、端末のキーボード最上段に PF キー (ファンクション・キー) 列の機能を解説した紙板があるので、その紙板をセットすること。かなり能率が上がる。またたま座った端末に FSO 用の紙板がない場合には、空いている他の端末から借りてくると良い。ここでは、各数字をそれぞれ決められたカラムに入力する必要があるデータのデータセット JAPAN.DATA を EDIT-FSO を用いて新規作成してみよう。

(1) EDIT コマンドの投入

```
READY
EDIT JAPAN DATA NONUM      [←]      (新規作成であるが、NEW の省略が可能)
KEQ52320I DATA SET NOT FOUND, ASSUMED TO BE NEW
INPUT
_ [←]      (データの投入を促しているが、なにも入力せず ENTER キーを押す)
```

(2) 入力モードの終了

ENTER キーが押されると、入力モードが終了して、EDIT 状態のサブコマンドモードにもどる。もどったら、FS サブコマンドを入力して EDIT-FSO を開始させる。

(3) FS サブコマンドの投入

FS サブコマンドは EDIT-FSO を呼び出すサブコマンドである。

```
EDIT
FS      [←]
```

(4) 空データセットの状態の EDIT-FSO 画面

これが EDIT-FSO 画面である。

```
EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==>   PAGE           COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
.....
*** END OF DATA SET ***
```

データが表示されるべき先頭行に 行番号 と空白行のみが表示され、データセットが空であることを示している。次に、データを入力するスペースを確保するために、4-3 に述べる行サブコマンド I を用いる。カーソル移動キーを使って行番号の先頭位置にカーソルを移動させる。

(5) I コマンドの投入 (I5 : 5 行分のスペースを確保する)

```
EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==>   PAGE           COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
I5...
*** END OF DATA SET ***
[←]
```

```

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
00100
00200
00300
00400
00500
*** END OF DATA SET ***

```

のように 5 行のデータを挿入できるスペースが生じた。

(6) 5 行のうち 1 - 70 カラムのデータのタイプ
 あとはこの画面上で入力すべき位置にキーボード右下のカーソル移動キーによってカーソルを移動させ、数字をタイプする。数字と数字の間の必要な空白はスペースキーを押す。
 実際の EDIT-FSO 画面では後述のように 70 カラムまでしか表示されないで、とりあえず 5 行のデータを 70 カラムまでタイプする。

(7) データの送信のために ENTER キーを押す
 タイプしたデータを計算機本体に送信するために ENTER キーを押す。すると、この 5 行のデータは画面上で輝度が強くなり計算機本体に送信されたことを知らせてくれる。

```

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4- ... 6----+----7--
00100 01100 1401757.0 1240613.0 1010177.0 ..... 487446.0
00200 01202 320154.0 307453.0 292286.0 ..... 267945.0
00300 01203 180728.0 184406.0 191802.0 ..... 131340.0
00400 01204 352619.0 320526.0 297189.0 ..... 214479.0
00500 01205 150199.0 158715.0 162059.0 ..... 123523.0
*** END OF DATA SET ***

```

[←]

5 行のデータの 70 カラム分のデータ入力が終了したら、残りの 71 - 80 カラム分のデータを入力しなければならない。

(8) PF11 キーを押す
 そのためには、PF11 キーを押して EDIT-FSO 画面を左に移動させて 71 - 80 カラムのスペースを確保する。PF11 キーを押すと次の画面を得る。

```

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> 2----+----3----+----4- ... 6----+----7----+----8----+----9-
00100 1240613.0 1010177.0 ..... 487446.0
00200 307453.0 292286.0 ..... 267945.0
00300 184406.0 191802.0 ..... 131340.0
00400 320526.0 297189.0 ..... 214479.0
00500 158715.0 162059.0 ..... 123523.0
*** END OF DATA SET ***

```

(9) 71 - 80 カラムのデータのタイプと ENTER キーを押す
 71 - 80 カラムの位置に入力すべきデータをタイプし、タイプし終わったら ENTER キーを押してデータを計算機に送信する。

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA

==>

```
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> 2-----3-----4-....6-----7-----8-----9-
00100 1240613.0 1010177.0 ..... 487446.0 393770.0
00200 307453.0 292286.0 ..... 267945.0 252756.0
00300 184406.0 191802.0 ..... 131340.0 186445.0
00400 320526.0 297189.0 ..... 214479.0 184242.0
00500 158715.0 162059.0 ..... 123523.0 110443.0
*** END OF DATA SET ***
```

[←]

(10) PF10 キーを押す

さらに 6 行以降のデータを入力するために、まず 1 - 70 カラムが表示される画面に戻すために、PF10 キーを押す。すると画面が下のように変化する。この画面でも 71 - 80 カラムに入力されたデータの一部が表示されている。

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA

==>

```
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> -----1-----2-----3-----4-....6-----7--
00100 01100 1401757.0 1240613.0 1010177.0 ..... 487446.0 39
00200 01202 320154.0 307453.0 292286.0 ..... 267945.0 25
00300 01203 180728.0 184406.0 191802.0 ..... 131340.0 18
00400 01204 352619.0 320526.0 297189.0 ..... 214479.0 18
00500 01205 150199.0 158715.0 162059.0 ..... 123523.0 11
*** END OF DATA SET ***
```

(11) (5) - (10) の処理を繰り返す

次に 6 行以降のデータを入力すべき行を確保するために、再び EDIT-FSO 画面の最終行の左端に行サブコマンド 15 をタイプし、ENTER キーを押す。

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA

==>

```
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> -----1-----2-----3-----4-....6-----7--
00100 01100 1401757.0 1240613.0 1010177.0 ..... 487446.0 39
00200 01202 320154.0 307453.0 292286.0 ..... 267945.0 25
00300 01203 180728.0 184406.0 191802.0 ..... 131340.0 18
00400 01204 352619.0 320526.0 297189.0 ..... 214479.0 18
15500 01205 150199.0 158715.0 162059.0 ..... 123523.0 11
*** END OF DATA SET ***
```

[←]

EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA

==>

```
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> -----1-----2-----3-----4-....6-----7--
00100 01100 1401757.0 1240613.0 1010177.0 ..... 487446.0 39
00200 01202 320154.0 307453.0 292286.0 ..... 267945.0 25
00300 01203 180728.0 184406.0 191802.0 ..... 131340.0 18
00400 01204 352619.0 320526.0 297189.0 ..... 214479.0 18
00500 01205 150199.0 158715.0 162059.0 ..... 123523.0 11
00600
00700
00800
00900
01000
*** END OF DATA SET ***
```

新しく 5 行分のスペースが得られたら、先程と同様に以下の処理を繰り返す

- 70 カラムまでのデータをタイプし ENTER キーを押して、計算機にデータを送信する。
- PF11 キーを押して 71 - 80 カラムが表示される画面を得る。
- 71 - 80 カラムの位置にデータをタイプし、ENTER キーを押す。
- PF10 キーを押して 1 - 70 カラムの画面に戻す。

(12) 現在の画面で拡張すべき行スペースがない場合

EDIT-FSO 画面に追加すべき行のスペースがなくなってしまったら、PF8 キーをもちいて次ページの画面を得るか、LOC サブコマンドを用いて表示開始行をずらす必要がある

(13) PF8 キーを押す

ここでは、PF8 キーを使って次ページを呼び出す方法を用いる。

PF8 キーを押すと下の画面を得る。データセットの最終行が EDIT-FSO 画面の先頭行に表示されている。

```
EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==>
ROW SCROLL ==> PAGE COLUMN SCROWLL ==> 40
=COLS> -----1-----2-----3-----4- .... 6-----7--
00500 01205 150199.0 158715.0 162059.0 ..... 123523.0 11
*** END OF DATA SET ***
```

(14) 行サブコマンド I の投入して (5)-(11) の処理を再び繰り返す

このように EDIT-FSO によるフルスクリーン形式におけるデータセットの新規作成は非常に便利な方法である。ただし、71 - 80 カラムへのデータ入力には PF10・PF11 キーを用いて画面を移動させなければならないので、いささか厄介である。この場合には、入力時のみ 1 - 70 カラムを使用して、その後で書換えプログラムを作成して 1 - 80 カラムを用いたデータセットに変換した方がよい。この方法には 5-4 「計算結果のデータセットへの出力」を参照すること。

(15) 入力したデータの保存 - 1 -

PF24(12) キーを押す

では、この入力したデータをデータセットに保存しよう。EDIT-FSO 画面の左上に [=]>]で示されるフィールドがあり、「サブコマンド」を入力することができる。このフィールドへカーソルを移動させるために、[PF24] (PF12 でもよい) キーを押す。

(16) 入力したデータの保存 - 2 -

SAVE サブコマンドの投入

```
EDIT-FSO(V01/L06) --- F1234.JAPAN.DATA
==> SAVE [←]
ROW SCROLL ==> PAGE COLUMN SCROWLL ==> 40
=COLS> -----1-----2-----3-----4- .... 6-----7--
00100 01100 1401757.0 1240613.0 1010177.0 ..... 487446.0 39
00200 01202 320154.0 307453.0 292286.0 ..... 267945.0 25
00300 01203 180728.0 184406.0 191802.0 ..... 131340.0 18
00400 01204 352619.0 320526.0 297189.0 ..... 214479.0 18
00500 01205 150199.0 158715.0 162059.0 ..... 123523.0 11
00600 01206 214694.0 206840.0 191948.0 ..... 119536.0 93
00700 01207 153861.0 141774.0 131568.0 ..... 92442.0 67
00800 01208 102915.0 91519.0 82727.0 ..... 59437.0 52
00900 01209 41715.0 50131.0 69871.0 ..... 107244.0 99
01000 01210 78311.0 72305.0 68712.0 ..... 55774.0 47
*** END OF DATA SET ***
```

(17) データセット JAPAN.DATA に保存された旨のメッセージと EDIT の終了

KEQ52460I SAVED DATA SET IN 'F1234.JAPAN.DATA'

EDIT (サブコマンドモード)
END [←] (END サブコマンドの投入)
READY

SAVE サブコマンドを使ってはじめて入力したデータが恒久的に修正されたことになる。SAVE サブコマンドを投入すると EDIT-FSO が終了し EDIT コマンドモードへ戻ってしまう。そのため、さらにデータの投入を続ける場合には、再び FS サブコマンドを入力しなければならない。これ以降は厳密にいうとデータセットの修正・追加になるので 4-2 「EDIT-FSO によるデータセットの修正・追加」を参照すること。

- PF キーの説明
- [PF7] キーは一つ前のページ画面にもどる
 - [PF8] キーは一つ後のページ画面にもどる
 - [PF10] キーは画面を左へ移動させる
 - [PF11] キーは画面を右へ移動させる
(70 カラムよりも大きい部分を見る)
 - [PF24(12)] キーはカーソルをサブコマンド行に移動させる

PF キーを押せば、入力したデータを計算機に送信する機能を果たす ENTER キーを押す働きがあるので、PF キーを押す前に ENTER キーを押す必要はない。

4-2 EDIT-FSO によるデータセットの修正・追加

修正及び追加は、既成のデータセットに対して行なうものなので、基本的な操作は新規作成の場合と同様である。EDITコマンドで既成のデータセットを呼び出す際には、もともとの形式が NONUM 形式になっていても、NONUM 指定は不要となる。さらに EDIT 状態のサブコマンドモードとしてから、FS というサブコマンドを入力する。これにより、画面の形式が変わってカラムの固定した画面になるので編集しやすくなるはずである。

前述のようにEDIT-FSO を用いてデータセットを編集する場合には、端末のキーボード上に PF キー列の機能を解説した紙板があるので、その紙板をセットすること。

手順

- (1) READY
EDIT データセット名 内容識別修飾子 [←]

既に保存(「カタログ」という)されたデータセットを EDIT コマンドで呼び出す場合には、データセットが「行番号付き」・「行番号なし」の如何にかかわらず、NONUM オペランドの指定は不要である。

指定したデータセットが NONUM 形式の場合には、次のようなメッセージが表示され、EDIT 状態のサブコマンドモードに入る。

- (2) KEQ52338I DATA SET データセット名.内容識別修飾子
NOT LINE NUMBER, NONUM ASSUMED
- (3) EDIT
FS [←]
- (4) EDIT-FSO 画面を得る。

新規作成および修正の場合にも、以上の手順は全く同じである。ただし、FS サブコマンドを入力しても直ぐに EDIT-FSO 画面が得られず、*** が表示されることがある。その場合には、もう一度 ENTER キーを押すと、EDIT-FSO 画面が表示される。

*** [←]

以下、EDIT-FSO 画面上で追加・修正が自由に行える。

- プログラムのデータセットの場合
これまで説明した (1)-(4) までの手順とほとんど同じであるが、FORTRAN77 プログラムのデータセットの場合には(例にデータセット名 MEAN.FORT77 とする)、

(1) READY
EDIT MEAN FORT77(FIXED) [←]

と指定しなければならない。FIXED オペランドはこのプログラムのデータセットが固定形式（正式には、標準形式と言う）で構成されていることを示している。形式にはこの他、FORT77 (FREE) と指定する自由形式もあるが、データセットの形式自体が可変長レコード形式になり、フロッピーディスク（IBM 形式）に格納できなくなるので、FIXED 形式の方がよい。

ただし、FIXED または FREE のいずれも指定しなかった場合には、自動的に FREE 形式になるので、特に新規作成の際には FIXED の指定を忘れないようにすること。

既存の FIXED 形式のプログラムのデータセットを呼び出す場合には、FIXED の指定を忘れると、FREE 形式扱いとなり実際のプログラムのデータセットの形式と合致しないので次のような警告メッセージがでて、EDIT コマンドの実行が中断され READY 状態に戻る。

KEQ52343I VARIABLE RECORD FORMAT REQUIRED FOR
FORT/FORT77(FREE) DATA SET+

READY

警告メッセージの末尾に「+」がついているので、さらに詳しくコンピュータからの情報を呼びだしてみよう。

READY

?

[←] (警告メッセージの末尾に「+」がついている場合に
さらに詳しく問い合わせるためのコマンド)

KEQ52343I EDIT AGAIN WITH FORT/PORT77(FIXED) OR USE CONVERT COMMAND

READY

この場合には FORT77(FIXED) を忘れないようにして、(1) をやりなおすこと。

また、EDIT コマンド投入時に プログラムのデータセットの場合には、行番号付きで処理されるため、NONUM オペランドを指定しても無視される。

これが (4) の EDIT-FSO 画面である。

```
EDIT-FSO(V01/L06) --- F1234.MEAN.FORT77
==>
ROW SCROLL ==> PAGE COLUMN SCROWLL ==> 40
=COLS> -----1-----2-----3-----4-----5
000010 * CALCULATION OF MEAN
000020 DEMENSION SCORE(100)
000030 READ(5,*) NUMBER
000040 DO 100 I=1,NUMBER
000050 READ(5,*) SCORE(I)
000060 100 CONTINUE
000070 SUM=0.0
000080 DO 200 I=1,NUMBER
000090 SUM=SUM+SCORE(I)
000100 200 CONTINUE
000110 XMEAN=SUM/NUMBER
000120 WRITE(6,*) XMEAN
000130 STOP
000140 END
*** END OF DATA SET ***
```

この画面をよく見ると、

```
000020 DEMENSION SCORE(100)
000110 XMEAN=SUM/NUMBER
```

の下線で示したスペルが間違っている。画面上のその位置にカーソルをあわせ、正しい文字に修正する。画面上のすべての間違いの修正が終わったら、画面が下の状態になり、

```

EDIT-FSO(V01/L06) --- F1234.MEAN.FORT77
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
000010 * CALCULATION OF MEAN
000020 DIMENSION SCORE(100)
000030 READ(5,*) NUMBER
000040 DO 100 I=1,NUMBER
000050 READ(5,*) SCORE(I)
000060 100 CONTINUE
000070 SUM=0.0
000080 DO 200 I=1,NUMBER
000090 SUM=SUM+SCORE(I)
000100 200 CONTINUE
000110 XMEAN=SUM/NUMBER
000120 WRITE(6,*) XMEAN
000130 STOP
000140 END
*** END OF DATA SET ***

```

続いて、ENTER キーを押す。この時のカーソルの位置はどこにあってもよい。その結果、この画面のすべての修正した部分の情報が計算機に送られ、仮の状態での修正が終了したことになる。ただし、まだ修正部分が永続的に修正されたわけではない。なぜならば、修正部分はデータセットに保存されていないからである。

EDIT-FSO 画面においても、DEL キーと INSERT キーは自由に使用できる。しかしながら、INSERT モードで何文字が挿入していると、それ以上文字が挿入できなくなる状態が発生する。これは、データセットのレコード形式が固定長形式の場合に、その行では右端（1 RECORD 80 文字）まで空白が詰められていて、新たに文字を挿入する余地がないからである。その場合にはその行の右端のいくつかの空白を ERASE EOF キーを押して取り去ってやればよい。しかし、この処理で空白を取り去っても画面の上では何等の変化もない。

では、この修正部分をデータセットに保存しよう。EDIT-FSO 画面の左上に [=>] 位置にサブコマンドを入力することができるフィールド（サブコマンド行）がある。このフィールドへカーソルを移動させるために、[PF24(12)] キーを押す。

(5) SAVE サブコマンドを使用して、修正部分を保存する

```

EDIT-FSO(V01/L06) --- F1234.MEAN.FORT77
==> SAVE [←]
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
000010 * CALCULATION OF MEAN
000020 DIMENSION SCORE(100)
000030 READ(5,*) NUMBER
000040 DO 100 I=1,NUMBER
000050 READ(5,*) SCORE(I)
000060 100 CONTINUE
000070 SUM=0.0
000080 DO 200 I=1,NUMBER
000090 SUM=SUM+SCORE(I)
000100 200 CONTINUE
000110 XMEAN=SUM/NUMBER
000120 WRITE(6,*) XMEAN
000130 STOP
000140 END
*** END OF DATA SET ***

```

(6) KEQ52460I SAVED IN DATA SET 'F1234.MEAN FORT77'

```

EDIT (EDIT 状態のサブコマンドモード)
END [←]

```

READY

SAVE サブコマンドを使ってはじめて修正部分が永続的に修正されたことになる。

修正した内容を SAVE [←] したら、一旦 EDIT-FSO 画面が消えて EDIT 状態へ戻ってしまう。そのため、再び修正を続けるためには、再び FS サブコマンドを入力しなければならない（その手順は (3) と同様である）。

FS サブコマンドで得られる画面はデータセットの最初のページである。例のような短いデータセットの場合には問題はないが、長いデータセットの場合には端末のキーボード上の [PF7] キーと [PF8] キーを使用して、ページめくりの要領ですべてのデータセットのページを検索しなければならない。

- [PF7] キーは一つ前のページ画面にもどる
- [PF8] キーは一つ後のページ画面にもどる

(7) LOC サブコマンドのタイプと ENTER キーを押す

データセットを編集する場合には、一度に見たい部分が PF7・PF8 キーの操作でうまく得られない場合がある。この場合には、その部分の先頭を EDIT-FSO 画面の先頭行として表示させる LOC サブコマンドがある。

カーソルがサブコマンド行になかったら PF24(12) キーを押す。するとカーソルがサブコマンド行に移動して、サブコマンドが入力可能になる。

```
EDIT-FSO (V01/L06) --- F1234.MEAN.FORT77
==> LOC 70      [←]
ROW SCROLL ==>  PAGE                COLUMN SCROLL ==> 40
=COLS> -----1-----2-----3-----4-----5
000010 *      CALCULATION OF MEAN
000020      DIMENSION SCORE(100)
000030      READ(5,*) NUMBER
000040      DO 100 I=1,NUMBER
000050      READ(5,*) SCORE(I)
000060 100 CONTINUE
000070      SUM=0.0
000080      DO 200 I=1,NUMBER
000090      SUM=SUM+SCORE(I)
000100 200 CONTINUE
000110      XMEAN=SUM/NUMBER
000120      WRITE(6,*) XMEAN
000130      STOP
000140      END
*** END OF DATA SET ***
```

```
EDIT-FSO (V01/L06) --- F1234.MEAN.FORT77
==>
ROW SCROLL ==>  PAGE                COLUMN SCROLL ==> 40
=COLS> -----1-----2-----3-----4-----5
000070      SUM=0.0
000080      DO 200 I=1,NUMBER
000090      SUM=SUM+SCORE(I)
000100 200 CONTINUE
000110      XMEAN=SUM/NUMBER
000120      WRITE(6,*) XMEAN
000130      STOP
000140      END
*** END OF DATA SET ***
```

このように表示開始行が 70 行になった画面が得られる。

EDIT-FSO 画面を終了したい時には、データセットを保存する場合には「SAVE サブコマンド」、保存しない場合には端末のキーボード上の [PF3] キーを押すと EDIT 状態のサブコ

マンドモードに戻る。続いて、END サブコマンド (ただし NO オペランドとともに) を入力すると READY 状態へ戻る。

○ EDIT コマンド投入時において指定するデータセット名の間違った場合

既に作成されているデータセットの修正の際に注意しなければならないのは、EDIT コマンドとともにタイプするデータセット名の間違った場合である。この場合には、NEW オペランドが無くても、既に保存されている (カタログされている) データセットリストの中にその名前がない場合には、新規作成のデータセットと判断されてしまう。以下にその際の処理手順を示す。

手順

(1) READY
EDIT NEAN FORT77 (FIXED) [←]

本来ならば、MEAN.FORT77 を指定すべきところを NEAN.FORT77 と呼び出してしまった。

(2) EDIT
KEQ52320I DATA SET NOT FOUND, ASSUMED TO BE NEW
INPUT
00010 - [←]
EDIT
END NO [←]
READY

このままでは別の NEAN.FORT77 というデータセットが作成されてしまうので、すぐに空行のまま ENTER キーを押し、EDIT 状態に戻ったら END NO サブコマンドを投入して、間違ったデータセット名の下での EDIT 状態を終了させる。

なお、この節で説明した修正ではいずれもミスタイプの場合のみであった。入力すべきプログラムのある行を失念して入力しなかったり、途中で入力すべきプログラム行に気がついたなどの場合の修正は、次の 4-3 「EDIT-FSO の行サブコマンド」と 4-4 「EDIT サブコマンドによる編集」で述べるので、そちらを参照すること。

4-3 EDIT-FSO の行サブコマンド

行サブコマンドは EDIT-FSO 画面の行番号 (NONUM の場合でも自動的にレコード番号が左端に表示される) の左端の位置に入力して、行の挿入・削除・複写・移動を行なうためのものである。この行サブコマンドがフルスクリーン形式の編集において最も力を発揮する。

○ I [n] ... 任意の連続 n 行のスペースの挿入 (n=1 の時、1 を省略できる)
(INSERT)

プログラム行・データ行の追加に用いる。ただし、複数の行のスペースを挿入しても、PF キーや [←] キーを押した時点で未入力行 (スペースを含む何らかの文字も入力されていない行) があった場合には、その未入力行は自動的に消えてしまうので注意すること。

<u>EDIT-FSO 画面</u>				
0010 A=1		1210 A=1 [←]		0010 A=1
0020 B=1 ==>	0020 B=1		==>	0011 [←]
				0012 未入力
				0020 B=1
	0010 A=1			
	0011 X=1			
==>	0020 B=1			

ただし、行番号の間隔が 10 きざみの場合には、I サブコマンドを使用して新しく 10 以上の行を挿入することができない。なぜならば、新しく確保した行スペースに行番号を割り振ることができないからである。

この場合にはまず 5 行挿入した後、PF24(12) キーを押してカーソルをサブコマンド行に戻し、RENUM サブコマンド (行サブコマンドではない) を入力して ENTER キーを押して、行番号を 10 きざみに付け換えて間隔を確保した後、再び I 行サブコマンドを使用して 5 行の

スペースを確保する。他の C・M 行サブコマンドの際にも同様である。

- D ... 1行分の削除
DD~DD ... 連続行の削除
(Delete)

2つの DD を削除したい部分の先頭行と末尾行に入力する。この2つの DD は同一画面上になくてもかまわない。従って、PF7・PF8 (画面の前後移動) を用いて別々の画面に入力すればよい。

```
0010 A=1      0010 A=1  [←]      0010 A=1
0020 B=1  ==> 0020 B=1      ==> 0030 C=1
0030 C=1      0030 C=1
```

```
0010 A=1      0010 A=1  [←]      0010 A=1
0020 B=1      DD20 B=1      ==> 0050 Y=1
0030 C=1      0030 C=1
0040 X=1  ==> DD40 X=1
0050 Y=1      0050 Y=1
```

- C-A 又は C-B ... 1行分の複写
CC~CC-A 又は CC~CC-B ... 連続行の複写
(Copy)

2つの CC を複写したい部分の先頭行と末尾行に入力する。この2つの CC は同一画面上になくてもかまわない。従って、PF7・PF8 (画面の前後移動) を用いて別々の画面に入力すればよい。

A (After) か B (Before) のどちらかの文字で複写する位置を指定しなければならない。A ならばその行の直後に、B ならば直前に複写される。位置指定をする時も必ずしも同一画面上でなくてもよい。

ここでは、C・CC・A・B を上手に組み合わせて、行単位の複写を実行するわけであるが、それぞれを入力すべき画面が別々の場合には、PF7・PF8 (画面の前後移動) を用いてめざす画面探し、順次に C・CC・A・B を入力すればよい。その中途の段階では、サブコマンドが実行保留になっていることを示す次のメッセージが出力される。

```
EDIT-FSO(V01/L06) --- F1234.MEAN.FORT77
==>
ROW SCROLL ==> PAGE          COLUMN SCROLL ==> 40
=COLS> KEQ57858I SUBCOMMAND EXECUTION PENDING -4-----5
000010 *          CALCULATION OF MEAN
000020          DIMENSION SCORE (100)
000030          READ(5,*) NUMBER
.....
```

以下にサブコマンドの使用例を示す。

```
0010 A=1      0010 A=1      0010 A=1
0020 B=1  ==> 0020 B=1      ==> 0020 B=1
0030 C=1      0030 C=1  [←]  0021 B=1
                                0030 C=1
```

```
0010 A=1      0010 A=1  [←]      0010 A=1
0020 B=1  ==> 0020 B=1      ==> 0020 B=1
0030 C=1      0030 C=1      0030 C=1
0040 X=1      0040 X=1      0040 X=1
0050 Y=1      0050 Y=1      0050 Y=1
                                0051 B=1
                                0052 C=1
                                0053 X=1
```

- M-A 又は M-B … 1行分の移動
- MM~MM-A 又は MM~MM-B … 連続行の移動 (Move)

2つの MM を移動したい部分の先頭行と末尾行に入力する。この2つの MM は同一画面上になくてもかまわない。従って、PF7・PF8 (画面の前後移動) を用いて別々の画面に入力すればよい。

A (After) か B (Before) のどちらかの文字で移動する位置を指定しなければならない。A ならばその行の直後に、B ならば直前に移動される。位置指定をする時も必ずしも同一画面上でなくてもよい。

```

0010 A=1      0010 A=1      0010 A=1
0020 B=1  ==> MM20 B=1  [←]  0050 Y=1
0030 C=1      0030 C=1      0051 B=1
0040 X=1      MM40 X=1      ==> 0052 C=1
0050 Y=1      0050 Y=1      0053 X=1
0060 Z=1      B060 Z=1      0060 Z=1
  
```

以上がよく用いる行サブコマンドである。

4-4 EDIT サブコマンドによる編集

EDIT-FSO 画面においては、上から2行目の左端にある ==> (サブコマンド行と呼ばれる) の位置に、必要なサブコマンドを入力して [←] とする。サブコマンドによっては、「サブコマンド 行番号」の形式で行番号の指定が必要なこともあるが、NONUM のデータセットの時には計算機が自動的に表示してくれるレコード番号を用いればよい。

- ==> SAVE … 作成したデータセットを保存する
- ==> END … EDIT 状態のサブコマンドモードを終了する
- ==> RUN … プログラムの翻訳・実行を行なう
- ==> LOCATE 行番号 … 指定した行番号のレコードを画面の先頭行として表示させる (LOC でも可)
- ==> TOP … データセットの開始行を画面の先頭行として表示させる
- ==> BOTTOM … データセットの最終行を画面の先頭行として表示させる (B でも可)
- ==> RENUM 新行番号の初期 (開始) 値 新行番号の増分 … 行番号の変更をしたい時に用いる。RENUM のみを入力したときには、全行番号を対象とし、10・20・30 … となる。

RENUM サブコマンドは NONUM 形式のデータセットに対して新規に行番号を付ける際にも用いる。その際には、73 - 80 カラムに行番号が格納されるのでその位置にデータが存在しないことをくれぐれも確認すること。

RENUM サブコマンドを実行すると、下記のように、73 - 80 カラムにあるデータが失われてもよいか、と計算機は確認を求めてくるので、ENTER キーを押す。

```

==> RENUM [←]
KEQ52569I WARNING, RENUMBERING DATA SET MAY CAUSE LOSS OF DATA+
HIT CARRIAGE RETURN TO RENUMBER, OR ENTER A NEW SUBCOMMAND
FS [←] (実行する場合、ENTER キーのみを押す)
FS [←] (取り止める場合、たとえば EDIT-FSO 画面に戻す)
  
```

- ==> UNNUM … 73-80 カラムの位置に格納されている行番号をとり去る。このサブコマンドが実際には行番号のないデータセットに実行された場合には、この位置のデータが失われるので注意すること。
- ==> MERGE 他のデータセット名、内容識別修飾子 … MERGE サブコマンドは、現在作成中のデータセットの途中に他のデータセットを組み込む場合に用いる。この場合、画面の先頭行の

直前に指定した他のデータセットが組み込まれる。

- ==> FIND 文字列
… 画面上で、特定の文字列を探し出すのに用いる。指定したものと同一の文字列は、他の文字よりも高輝度で表示される
- ==> CHANGE 行番号1 行番号2 '旧文字列' '新文字列' ALL
… ある文字列を別の文字列に置換する。
行番号付きのデータセットの場合には、例のように行番号1と行番号2によって、置換する範囲を指定する
行番号なしのデータセットの場合には、LOCATE コマンドで置換する先頭の行を画面表示の先頭行にしておき（置換する先頭行に行指標を移動させる）、行番号1と行番号2の代わりに
行番号1 → *（行指標が現在指している行）
行番号2 → 行指標から何番目の行かを示す数字
を使用して、置換の範囲を指定する。
オペランド ALL を追加すると、変更対象範囲内の指定した文字列がすべて置換される。ALL を指定しないと、変更対象範囲内の各行の最初の指定文字列しか置換されない

EDIT コマンドモードのサブコマンドもすべて、==> の右側に入力することができる。
また、次のように X とともにコマンドをタイプする方法で、コマンド（もちろん EDIT を除く）も入力することができる。

==> X ALLOC

ALLOC にはサブコマンドもあるが、コマンドとしての使用例である。
コマンドの中には、EDIT コマンドモードのサブコマンドと同じ名前のものであり、同様な働きをすることが多い。しかし、両者は意味が異なるので注意が必要である。例えば、MERGE コマンドは作成済みの2個のデータセットを結合するものであるが、EDIT 状態での MERGE サブコマンドは、データセットの作成の途中で他のデータセットを組み込むために用いる。

4-5 PFキーの配列

EDIT-FSO 画面で用いる場合の例を挙げておこう。

- 【 PF 1 】 HELP機能（コマンド一覧表の表示など）
【 PF 2 】
【 PF 3 】 FSO操作の終了
次の画面は
EDIT
となり、EDIT-FSO 画面は消えてしまう。
- 【 PF 7 】 画面を1つ前のページに戻す（行番号小へ）
【 PF 8 】 画面を1つ後のページへ送る（行番号大へ）
【 PF 9 】
【 PF 10 】 画面を左へ移動
【 PF 11 】 画面を右へ移動（70 カラムよりも大きい部分をみる）
【 PF 12 】 カーソルをサブコマンド行に移動させる
- 【 PF 24 】 カーソルをサブコマンド行に移動させる

なお、PF キーの機能は SORP（会話型リモートバッチ処理ジョブの取りだし請求）コマンドの使用時などにおいても同様な機能を果たすので非常に有用である。

5 プログラムの実行 (EDITコマンドを用いる場合)

5-1 翻訳 (COMPILE) と実行

プログラムのデータセットを作成してしまったり、いよいよ実行の段階へ進むわけである。プログラムの実行は大まかに見て、二段階に分けられる。

大型計算機はパーソナルコンピュータ (BASIC 言語) と異なり、まず FORTRAN 言語のプログラム (正確にはソースプログラムと呼ばれる) を機械語に変換 (翻訳 COMPILE) する。そして、その機械語に変換されたもの (オブジェクト・モジュール) と必要なシステムライブラリーのデータセットなどをリンクして、一つの実行可能な形式 (ロードモジュール) に直した上で、プログラムによって指示された手続きを実行するのである。従って、実行すると、端末には翻訳の終了を示す「END OF COMPILATION」と処理結果が二段階で出力される。

従って、エラーも各段階で発生する可能性があるが、ほとんどのエラーは COMPILE 時と実行時の二種類ある。エラーが発生した場合には、どちらのものかよく注意する必要がある。会話型リモートバッチ処理には文法チェック専用の JOB があるので、これを使うと便利だが、COMPILE 時のエラーしか検出できない。この方法については、6 章で解説する。

RUN (EDIT 状態の時にプログラムの翻訳・実行を行なうサブコマンド) を中断したい時には、[ALT] キーと [DUP] キーを同時に押すことによって止めることができる。このような処理が必要になるのは、プログラム中に間違いがあって無限ループに入ってしまった場合である。この機能をアテンション割り込み機能という。

5-2 データをキーボードから入力して実行する場合

プログラムの実行は、EDIT 状態で RUN サブコマンドを用いる。したがって、プログラム中の READ 文の機番が 5 番 (入力装置としてキーボードが予め自動的に割り当てられている) の場合には、実行時にディスプレイに「_」と表示されるので、適宜データをキーボードから入力する。

EX.

- ```
(1) READY
 EDIT PROG FORT77(FIXED) [←]
(2) EDIT
 RUN [←]
(3) FORTRAN77 COMPILER ENTERED
 END OF COMPILATION
 00020 ? 行番号 20 の READ 文による入力促進記号
 - 2,5 2, 5 は入力データの例
 213,24321 546,54652 (出力された結果)
(4) END OF GO,SEVERITY CODE = 00
 EDIT
 END [←]
 READY
```

### 5-3 データをデータセットから入力して実行する場合

TSS を利用していると、使用するデータを予めデータセットとして作成・保存しておき、プログラムの実行時にデータセットから読み込んで計算する方法が便利である。例えば、一つのデータのデータセットを種々のプログラムによって何度も計算したり、大量なデータを様々なパラメータの下で計算する場合には、パラメータをキーボードから、大量データはデータセットから入力すると便利である。

ALLOC (サブ) コマンドを用いると、プログラム中の READ 文の機番ごとに、入力装置・入力データセットを自由に割り当てることができる。その際、プログラム中の READ 文と対になった FORMAT 文の書式と実際のデータとなるデータセットの書式はくれぐれもキチンと対応させておかなければならない。

では、データの個数をキーボードから、データをデータセット (POP.DATA に格納されている) から入力して、その平均値を求める下のプログラムを例に実行して見よう。

```
00010 * CALCULATION OF MEAN
```

```

00020 DIMENSION SCORE(100)
00030 READ(5,*) NUMBER
00040 DO 100 I=1,NUMBER
00050 READ(1,*) SCORE(I)
00060 100 CONTINUE
00070 SUM=0.0
00080 DO 200 I=1,NUMBER
00090 SUM=SUM+SCORE(I)
00100 200 CONTINUE
00110 XMEAN=SUM/NUMBER
00120 WRITE(6,*) XMEAN
00130 STOP
00140 END

```

このプログラムの READ 文を見ると、行番号 30 と 50 の二箇所にある。

すなわち、このプログラムでは NUMBER (データの個数) は機番 5 番の READ 文から、SCORE(I) (データ) は機番 1 番の READ 文から入力しようと意図している。機番 5 番は予め自動的にキーボードが割り当ててあるが、機番 1 は ALLOC (サブ) コマンドによって入力データセットを割り当てる。ALLOC コマンドの指定形式は、

```
ALLOC DA(入力するデータセット名.DATA) FILE(FT機番F001) SHR
```

のようにする。ただし FILE オペランド (正式にはデータセット定義名と呼ぶ) の機番は二桁で指定し、機番 1 番の場合には FT01 とする。

一方、行番号 120 にある WRITE 文の機番 6 番はディスプレイが予め自動的に割り当ててあり、XMEAN の値はディスプレイに出力されるはずである。

手順

```

READY
EDIT MEAN FORT77(FIXED) [←]
EDIT
ALLOC DA(POP.DATA) FILE(FT01F001) SHR [←]
EDIT (ALLOC は EDIT 状態で入力可能な
RUN [←] サブコマンドでもある)
FORTRAN77 COMPILER ENTERED
END OF COMPILATION

```

行番号 30 の READ 文 (機番 5 番) にもとづく入力促進記号「 00030 ?\_ 」が画面に表示されるので、以下データの個数を入力する。

```

00030 ?
_ 5 (データの個数) [←]

```

するとデータそのものは ALLOC コマンドによって行番号 50 の READ 文の機番 1 に割り当てられたデータセットから自動的に読み込まれる。

```

3.062
END OF GO,SEVERITY CODE = 00
EDIT

```

以上の要領でプログラムを実行することができる。さらに、同じプログラムを使って、別のデータ (データセット名 WORK.DATA) の平均値を求めて見よう。

```

EDIT
ALLOC DA(WORK.DATA) FILE(FT01F001) SHR [←]
KEQ56246I DATA SET 'F1234.WORK.DATA' NOT ALLOCATED, FILE IN USE
KEQ56112A ENTER 'FREE' OR 'END'+-

```

と表示され、機番 1 番はすでに割当てられてしまっており、使用中であることを示している。WORK.DATA のデータを使用して計算する場合には、POP.DATA の割当てを解除しなければなら

ない。これには、KEQ56112A の表示どおり FREE サブコマンドを入力しなければならない。

```
FREE DA(POP.DATA) [←]
または、
FREE FILE(FT01F001) [←]
EDIT
```

すると、WORK.DATA が機番 1 に割当られて、EDIT 状態のサブコマンドモードになる。以下、POP.DATA の場合と同様に RUN とタイプし、入力促進記号「 00030 ?\_ 」が画面に表示されるのでデータの個数を入力する、そして結果が出力される。

また、ALLOC (サブ) コマンドによるデータセットの再指定の際に、FILE IN USE の状態にならないようにするために、

```
ALLOC DA(WORK.DATA) FILE(FT01F001) SHR REUSE [←]
```

末尾に REUSE オペランドを付け加えてもよい。これはすでに割り当てられていたデータセットを解放して、新たに別のデータセットの割当てを可能にするオペランドである。

#### 5-4 計算結果のデータセットへの出力 (会話型処理の場合)

計算結果を画面に出力させただけでは、その結果を再びデータとして利用できない。それは WRITE 文中では出力装置が 6 番、つまりディスプレイに割り当てられているからである。

ところが、計算結果をデータセットに出力し、さらに他のプログラムを用いて別の処理のデータとして使用することは、非常に多い。このために、計算結果の出力先を ALLOC (サブ) コマンドによって適当なデータセットに変更することができる。

そのために、EDIT-PSO 画面の上で WRITE 文中の機番を 2 番に書き換え、ALLOC サブコマンドによって機番 2 に出力用データセットを割り当てておき、再実行させて出力された計算結果をそのデータセットとして保存させることができる。

こうしておけば、データセットに記録しておいた計算結果を後で別の処理に使うのも、ディスプレイや NLP に出力させるのも思いのままとなる。

では、POP2.DATA (データの個数・データの両方が書き込んである、とする) を入力用データとし、平均と偏差を求めて (プログラムのデータセット名 DEVT.FORT77)、それを DEV.DA TA というデータセット (新しく作成する) に保存してみよう。

```
00010 * CALCULATION OF DEVIATION
00020 DIMENSION SCORE(100)
00030 READ(1,*) NUMBER
00040 DO 100 I=1,NUMBER
00050 READ(1,*) SCORE(I)
00060 100 CONTINUE
00070 SUM=0.0
00080 DO 200 I=1,NUMBER
00090 SUM=SUM+SCORE(I)
00100 200 CONTINUE
00110 XMEAN=SUM/NUMBER
00120 WRITE(6,1000) ((SCORE(I)-XMEAN), I=1,NUMBER)
00130 1000 FORMAT(8F10.4)
00140 STOP
00150 END
```

手順

```
READY
EDIT DEVT FORT77(FIXED) [←]
EDIT
ALLOC DA(POP2.DATA) FILE(FT01F001) SHR [←] ... 入力用データセットの
EDIT 割り当て
RUN [←]
FORTRAN77 COMPILER ENTERED
END OF COMPILATION
```

```
3.0622 15.9634 10.2841 5.1420
END OF GO,SEVERITY CODE = 00
EDIT
```

のように、まず WRITE 文の機番を 6 番 (ディスプレイ) のままにして、計算結果が正しく出力されるかどうか、確認する。正しい結果が確認されたら、もう一度プログラムを実行させ、計算結果をデータセットに出力させる手続きをする。

```
EDIT
FS [←]
EDIT-FSO 画面上で、
00120 WRITE(6,1000) ((SCORE(I)-XMEAN,I=1,NUMBER) を
00120 WRITE(2,1000) ((SCORE(I)-XMEAN,I=1,NUMBER) に
```

修正し、DEVT.FORT77 に保存する (SAVE サブコマンド)。

修正・保存の終了後、ATTRIB・ALLOC サブコマンドをもちいて、DEV.DATA という出力用データセットを新規に作成し、その後もう一度 ALLOC サブコマンドを用いて、DEV.DATA を機番 2 に割り当てる。ただし、この時には改めて入力用データセット POP2.DATA を機番 1 に割当なくてもよい。

```
EDIT
ATTRIB PS1 BLKSIZE(3120) LRECL(80) DSORG(PS) RECFM(F B)
... PS1 の名前で新規作成データセットの属性を定義
```

```
EDIT
ALLOC DA(DEV.DATA) SPACE(10,10) TRACKS NEW RELEASE CATA USING(PS1)
... DEV.DATA (器のみ) の新規作成
```

```
EDIT
ALLOC DA(DEV.DATA) FILE(FT02F001) OLD
... 機番 2 (正式には、データセット定義名 FILE(FT02F001)) に
DEV.DATA を割り当て
```

```
EDIT
RUN [←]
FORTRAN77 COMPILER ENTERED
END OF COMPILATION
END OF GO,SEVERITY CODE = 00
EDIT
END [←]
```

DEVT.FORT77 の EDIT コマンドモードを終了させ、計算結果を書き込んだデータセットの内容を確認しなければならない。

```
READY
EDIT DEV DATA [←]
KEQ52338I DATA SET DEV,DATA NOT LINE NUMBER, NONUM ASSUMED
EDIT
```

となるから、EDIT-FSO 画面を呼出確認する。ただし、プログラムの実行中にエラーが生じた場合には、プログラムを修正した後に再度実行する前に、新規に作成してしまった出力用データセット DEV.DATA を削除しておかなければならない。

エラーが生じた場合の手続きについては、5-5 「コマンドプロシジャを用いたデータセットからの入出力」を参照すること。

## 5-5 コマンドプロシジャ (マクロ命令) を用いたデータセットからの入出力

5-3・5-4 において、データセットからの入出力の方法を説明した。このデータセットからの入出力を大型計算機を使用する際には、頻繁に行なう処理である。しかし、この手順は非常に複雑であるから、初心者にはなかなか難しい。このような処理を行なう際には、コマンドプロシジャ (富士通ではマクロ命令をコマンドプロシジャという) を用いるとよい。



コマンドプロシジャとは、予め必要なコマンド・サブコマンドを必要な順序に並べたものを、データセット（この内容識別修飾子は CLIST）として作成しておき、EXEC コマンドとともに用いて、一連の処理を自動的に行なう手法である。

我々が使用する課題に、データセットからの入出力のためのコマンドプロシジャのデータセットを、

```
COMMAND.CLIST(FILEREAD)
```

```
COMMAND.CLIST(FILEWR)
```

という、名前をつけて区分データセットを予め作成しておくことと便利である。

では、上記のコマンドプロシジャが作成してあるものとして、区分データセットの編集・実行のためのユーティリティコマンド MLIST コマンドをもちいて、実行してみよう。

```
READY
```

```
MLIST COMMAND [←]
```

```
==>
```

```
-- MLIST -- V20/L04 -- F1234.COMMAND.CLIST ----- ID(M1) -- 1/ 11
```

```
FILECR ==> -
```

```
FILEREAD ==> -
```

```
FILEWR ==> -
```

```
** END **
```

上のような COMMAND.CLIST の編集・処理のためのメニュー画面がでるので、EDIT コマンド・EXEC コマンドなどをタイプして、コマンドを実行することができる。

5-4 で説明した、データセットから入力し新規に作成したデータセットに計算結果を書き込む処理をコマンドプロシジャを用いて実行してみよう。

```
==>
```

```
-- MLIST -- V20/L04 -- F1234.COMMAND.CLIST ----- ID(M1) -- 1/ 11
```

```
FILECR ==> -
```

```
FILEREAD ==> -
```

```
FILEWR ==> EDIT [←] -
```

```
** END **
```

メニュー画面のコマンド投入フィールドに、EDIT コマンドをタイプし ENTER キーを押す。

```
KEQ2338I DATA SET COMMAND.CLIST(FILEWR) NOT LINE NUMBERED, NONUM ASSUMED
```

```

```

と表示されるので、もう一度 ENTER キーを押すと、EDIT-FSO 画面が表示される。以下はEDIT-FSO操作と全く同様に行なう。そして、処理を行なうプログラムのデータセット・入力するデータセット・出力用の新しく作成するデータセット名を所定の位置に正しくタイプし、SAVE すればよい。

```
EDIT-FSO(V01/L06) --- F1234.COMMAND.CLIST(FILEWR)
```

```
==> SAVE [←]
```

```
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
```

```
=COLS> -----1-----2-----3-----4-----5-----6
```

```
000010 FREE ALL
```

```
000020 ATTRIB PS1 BLKSIZE(3120) LRECL(80) DSORG(PS) RECFM(F B)
```

```
000030 ALLOC DA(DEV.DATA) SPACE(10,10) TRACKS NEW RELEASE CATA USING(PS1)
```

```
000040 ALLOC DA(POP2.DATA) FILE(FT01F001) SHR
```

```
000050 ALLOC DA(DEV.DATA) FILE(FT02F001) OLD
```

```
000060 EDIT DEVT.PORT77(FIXED)
```

```
000070 RUN
```

```
000080 END
```

```
000090 FREE ALL
```

```
*** END OF DATASET ***
```

上記の画面を得て、ENTER キーを押す。すると、COMMAND.CLIST(FILEWR) のデータセットに保存された旨のメッセージが表示された後、MLIST のメニュー画面に戻る。

```

==>
-- MLIST -- V20/L04 -- F1234.COMMAND.CLIST ----- ID(M1) -- 1/ 11
FILECR ==> -
FILEREAD ==> -
FILEWR ==> EXEC [←] -
** END **

```

FILEWR のコマンドフィールドにコマンドプロシジャを実行させる EXEC コマンドをタイプし、ENTER キーを押すと、指示した一連のコマンドが以下のようなメッセージが表示されながら、順序正しく実行される。

```

FORTRAN77 COMPILER ENTERED
END OF COMPILATION
END OF GO,SEVERITY CODE = 00
*** [←]

```

と表示されるので、ENTER キーを押すと、そして再び MLIST のメニュー画面に戻る。

```

==>
-- MLIST -- V20/L04 -- F1234.COMMAND.CLIST ----- ID(M1) -- 1/ 11
FILECR ==> -
FILEREAD ==> -
FILEWR ==> *EXEC ■ -
** END **

```

MLIST メニューを終了するには、PF3 キーを押すと READY 状態にもどる。ただし、プログラムの実行中にエラーが起ると

```

FORTRAN77 COMPILER ENTERED
END OF COMPILATION
JQB7200-U WORKING AREA OVERFLOWED
END OF GO,SEVERITY CODE = 16
EDIT

```

と表示され、一連の処理が EDIT 状態で止まってしまう。この場合には、まず入出力に割り当てたデータセットを解放し、EDIT 状態を終了させなければならない。

```

EDIT
FREE ALL [←] 割り当てたデータセットの解放
EDIT
END [←]

```

MLIST コマンドのメニュー画面に戻ったら、PF3 キーを押して READY 状態にする。そして、出力用データセットを削除してから、プログラムの修正を行なう。ただし、この例でのエラーは文法エラーではなくリージョンサイズの不足が原因なので、LOGOFF してセッションを終了させ、再度開設する。

```

MLIST のメニュー画面
READY
DEL DEV.DATA [←]
ENTRY (A) F1234.DEV.DATA DELETED
READY
LOGOFF [←]
セッション終了のメッセージ
JCE005 SYSTEM READY
LOGON ON TSS F1234 S(2048) [←]

```

コマンドプロシジャにはエラー処理をふくむ手続きを指示できるので、研究するとよい。藤村直美：「TSSによる情報処理」（朝倉書店）などが詳しい。

## 6 会話型リモートバッチ処理

### 6-1 会話型リモートバッチ処理とは

計算機の処理形態を大別すると、TSS 処理とバッチ処理の二つに分けられる。

TSS 処理は、複数の人間が一台の大型計算機をそれぞれの端末から同時に使用する形態である。したがって、その利用には CPU TIME、リージョンサイズ、会話時間などの制限があり、超大型の計算には適していない。

これに対して、バッチ処理では、必要なプログラム・データなどをある一定時間ためておいて一括処理をする。したがって、TSS 処理と違い人間が計算途中に介入する必要はなく、その間他の仕事をしていてもよい。また、バッチ処理における CPU TIME・リージョンサイズなどの制限値も大きく巨大な計算に適している。バッチ処理に適した処理は、

- (1) TSS 処理では多くのコマンドを使用する場合
- (2) 大量のデータを処理する場合
- (3) 実行時間が長い
- (4) バッチ処理専用のライブラリープログラムを使用する場合

などがあげられる。TSS によるバッチ処理を会話型リモートバッチ処理と呼ぶ。

### 6-2 会話型リモートバッチ処理用のジョブ制御文 (JCL)

バッチ処理では、これまで作成してきた TSS 処理のためのプログラム・データがそのまま使用できる。バッチ処理を行なうために新たに必要な作業は、バッチ処理用の一連の作業を命令するためのコマンドを並べたデータセットを作成することである。この一連の命令を並べたものをジョブ制御文 (略して JCL と称する) という。

同じ種類の処理には同じ JCL を用いることができるので、使用頻度の高いバッチ処理のために、JCL が予め CARD.CNTL という区分データセットに作成してあるとたいへん効率的である。したがって、JCL の作成では CARD.CNTL のなかから、これから処理しようとするメンバーを選び、実際に使用するプログラム・データのデータセット名に書きかえるだけでよい。JCL の内容については「センター利用の手引」を参照すること。

また、CARD.CNTL 以外にもライブラリープログラム SPSS・BMDP・ANALYST のバッチジョブ用 JCL を区分データセット SPSS.CNTL・BMDP.CNTL・ANALYST.CNTL に作成しておくとも良い。

では、ここでは CARD.CNTL という JCL 用の区分データセットが作成してあるものとして (3-4 「会話型リモートバッチ処理用の JCL の作成方法」)、MLIST コマンドを用いて CARD.CNTL のメンバーをみてみよう。

- (1) READY  
MLIST CARD [-]
- (2) 区分データセット CARD.CNTL のメンバーが表示される。

```
==>
-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
 BATCH ==> -
 COMPPS ==> -
 PRINTPS ==> -
 SUBCHK ==> -
 TAPERED==> -
 TAPEWR ==> -
** END **
```

以下に各メンバーの用途を示す。

- BATCH: 別々のデータセットとして格納してあるプログラム・データを用いて実行するバッチジョブ用の JCL
- COMPPS: 順データセットの濃縮印刷用バッチジョブの JCL
- PRINTPS: 順データセットの印刷用バッチジョブの JCL
- SUBCHK: データセットに格納してあるプログラムの文法チェック用 JCL
- TAPERED: 磁気テープ上のデータセットをホスト計算機の順データセットとして保存するための JCL
- TAPEWR: ホスト計算機の順データセットを磁気テープに格納するための JCL

(3) 上のような区分データセットの処理のためのメニュー画面があるので、EDIT コマンド、SUBMIT コマンドなどをタイプして、コマンドを実行することができる。

以下に、データセット MEAN.FORT77 に格納されたプログラム、データセット POP.DATA に格納されたデータを用いて実際にバッチ処理を行なってみよう。

まず、通常のバッチ処理用の JCL を呼び出して、使用するプログラム・データのデータセットに書きかえなければならない。メンバー名は BATCH であるからその行にカーソルを移動させ、コマンドフィールドに EDIT とタイプする。

```
-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> EDIT
COMPPS ==>
PRINTPS ==>
SUBCHK ==>
TAPEREAD==>
TAPEWR ==>
** END **
```

ENTER キーを押す。すると下のような画面が表示されるので、

```
KEQ2338I DATA SET CARD.CNTL(BATCH) NOT LINE NUMBERED, NONUM ASSUMED
*** [←]
```

もう一度 ENTER キーを押すと、EDIT-FSO 画面が表示される。以下はEDIT-FSO操作と全く同様にやう。

#### (4) JCL 上のデータセット名の変更

これが、通常のバッチ処理用の JCL である CARD.CNTL(BATCH) の内容である。

```
EDIT-FSO(V01/L06) --- F1234.CARD.CARL(BATCH)
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
000100 //F1234C JOB CLASS=A
000200 // EXEC FORT77,STEP=CLG
000300 //FORT.SYSIN DD DSN=F1234.AREA.FORT77,DISP=SHR
000400 //GO.FT05F001 DD DSN=F1234.JAPAN.DATA,DISP=SHR
000600 //
*** END OF DATASET ***
```

必要に応じて次のような変更を行なう。

- ジョブ名 C を自分の整理の必要上から A に変更する F1234A
- これから実行するジョブは巨大な計算なので、JOB CLASS を A クラスから C クラスに変更する CLASS=C  
メモリーサイズを標準値の 512KB から 1024KB まで拡大する REGION=1024K
- 実行するプログラム・データのデータセット名を指定する  
行番号300 にて指定されているプログラムのデータセット AREA.FORT77を  
MEAN.FORT77 に変更する。  
行番号400 で指定されているデータのデータセット JAPAN.DATA を  
POP.DATA に変更する。
- ここでは変更しないが、行番号 500 の FT05F001 はプログラムのなかのデータを  
読み込む READ 文の機番に対応させなければならない。

```
EDIT-FSO(V01/L06) --- F1234.CARD.CARL(BATCH)
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
000100 //F1234A JOB CLASS=C,REGION=1024K
000200 // EXEC FORT77,STEP=CLG
000300 //FORT.SYSIN DD DSN=F1234.MEAN.FORT77,DISP=SHR
```

```
000400 //GO.FT05F001 DD DSN=F1234.POP.DATA,DISP=SHR
000600 //
*** END OF DATASET ***
```

EDIT-FSO 画面上で新しいデータセット名などの変更をタイプし終わったら、ENTER キーを押した後、PF24 (12) キーを押して、カーソルをサブコマンド行に戻す。そして、SAVE サブコマンドを入力する。

```
EDIT-FSO(V01/L06) --- F1234.CARD.CARL(BATCH)
==> SAVE [←]
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40
=COLS> ----+----1----+----2----+----3----+----4----+----5
000100 //F1234A JOB CLASS=C,REGION=1024K
000200 // EXEC FORT77,STEP=CLG
000300 //FORT.SYSIN DD DSN=F1234.MEAN.FORT77,DISP=SHR
000400 //GO.FT05F001 DD DSN=F1234.POP.DATA,DISP=SHR
000600 //
*** END OF DATASET ***
```

すると、実行するための JCL が作成保存された旨のメッセージが表示される。再び、メニュー画面に戻る。

```
DATASET SAVED IN 'F1234.CARD.CNTL(BATCH)'
EDIT (EDIT 状態のサブコマンドモード)
END [←]
```

```
-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> *EDIT -
COMPPS ==> -
PRINTPS ==> -
SUBCHK ==> -
TAPERREAD==> -
TAPEWR ==> -
** END **
```

#### (5) SUBMIT コマンドの投入

SUBMIT コマンドはバッチ処理の実行を命令するコマンドである。このコマンドを、さきの EDIT コマンドの時と同様にコマンドフィールドにタイプし、ENTER キーを押す。

```
-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> SUBMIT [←]
COMPPS ==> -
PRINTPS ==> -
SUBCHK ==> -
TAPERREAD==> -
TAPEWR ==> -
** END **
```

すると次のようなバッチジョブが受け入れられた旨のメッセージが表示され、ENTER キーを押すと、また MLIST のメニュー画面に戻る。

```
*** F1234A : (RECEIVED) ***
*** F1234A (JOB6640) 1234 : (JOB ACCEPTED) *** FIB CN(01)
JOB F1234A(JOB06640) SUBMITTED

[←]
```

```
-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> *SUBMIT -
COMPPS ==> -
PRINTPS ==> -
SUBCHK ==> -
TAPERREAD==> -
TAPEWR ==> -
** END **
```

(6) MLIST メニュー画面の終了

MLIST メニューを終了するには、PF3 キーを押すと READY 状態にもどる。

### 6-3 バッチジョブのモニター

バッチジョブを依頼したままではその処理結果を手にすることはできない。したがって、バッチジョブの実行を推移をみるモニターとジョブが終了すれば取りだし請求をしなければならない。

(1) @ST コマンドによる全体のジョブ処理状況の把握

まず、バッチジョブが混雑しているかどうかを @ST コマンドで確かめる。もし、混雑はひどい場合には、さきほど依頼したバッチジョブをキャンセルして、可能な限りジョブクラスを下げる。バッチジョブはジョブクラスが低いものから優先的に処理される。

```
READY
@ST [←]
WAITING 2 JOBS ON B CLASS CN(11)
WAITING 1 JOBS ON C CLASS CN(11)
READY
```

待ちジョブがない場合には、

```
WAITING JOB IS NOTHING CN(11)
```

と表示される。

ただし、この @ST コマンド投入後、実行待ちジョブが表示されず READY 状態になってしまう場合がよく起る。これは、実行待ちジョブ数の表示に手間取っているためであり、他の処理をしている間に暫時表示される。

(2) ST コマンドによる自分のジョブ処理状況の把握

何度か @ST コマンドを行なって自分の依頼したジョブが実行されていそうと思ったら次に ST コマンドを投入する。

```
READY
ST [←]
KEQ56211I JOB F1234A(JOB06640) IS WAITING FOR EXECUTION
... 一番新しく依頼したバッチジョブは実行待ちである
KEQ56211I JOB F1234C(JOB06638) IS EXECUTING
... このジョブは実行中である
KEQ56211I JOB F1234#(TSU00774) IS EXECUTING
... TSS のセッションが実行中であることを示す
KEQ56211I JOB F1234X(JOB06625) IS WAITING FOR OUTPUT
KEQ56211I JOB F1234Z(JOB06621) IS WAITING FOR OUTPUT
... すでに処理が終了し、取りだし請求を待っている
```

```
READY
```

このように複数のバッチジョブが依頼されている場合には、すべてのジョブの状態が表示される。したがって、一つ一つのジョブネームは変えておいたほうがよい。ただし同じジョブネームでも JOB ナンバーによって区別することはできる。また、JOB ナンバーは番号が大きいほど後に依頼されたジョブである。

(3) MSO コマンドによるジョブ結果の取りだし請求  
処理の終了を ST コマンドによって確認した後は、取りだし請求を待っている F1234X・F1234Z の結果を取りだし手続きに進むことになる。  
その方法は、6-4 「SORP 機能 (バッチジョブの取りだし請求)」を参照のこと。

#### 6-4 SORP 機能

SORP 機能を利用すると、実際に用紙 (1 PAGE 66 行、1 行 132 COLUMN) に出力することなく、用紙に出力したのと同じ状態で画面に表示させることができる。必要のないジョブ結果は、画面上のみで見て用紙に出力させずに消去することもできる。

また、後述のように SORP 機能ではファンクションキーを用いるので、情報処理センターの端末には SORP 機能下でのファンクションキーの働きを解説した紙板があるのでこれを必ず用いること。もし、使用している端末にその紙板がない場合には使用されていない他の端末から借りるとよい。

ここで投入する MSO コマンドはジョブ結果の取りだし請求手続 SORP 機能を呼び出すコマンドである。複数のバッチジョブの実行が終了していても、簡単に望むジョブを取り出せる便利なコマンドである。

##### (1) MSO コマンドの投入

```
READY
MSO [←]
```

すると、SORP のメニュー画面が表示される。

##### (2) SORP の初期メニュー画面

```
-----< MENU FOR SORP >-----
YOUR JOBS ARE AS FOLLOWS.
 (SEQ-NO.) (JOB-NAME)
 1 F1234X(JOB04425)
 2 F1234Z(JOB04421)

JOB SELECTION:
 SEQ-NO. ==> 1

DSPRINT(PF5) PICKED UP PAGE:
 DATASET ==>
```

##### (3) SORP 初期メニュー画面上での取りだしジョブの選択

取り出すべきジョブは F1234Z なので、メニュー画面上の JOB SELECTION: SEQ-NO. の指定を 1 から 2 に変更しなければならない。カーソルをその位置に合せて 2 とタイプする。

```
-----< MENU FOR SORP >-----
YOUR JOBS ARE AS FOLLOWS.
 (SEQ-NO.) (JOB-NAME)
 1 F1234X(JOB04425)
 2 F1234Z(JOB04421)

JOB SELECTION:
 SEQ-NO. ==> 2 [←]

DSPRINT(PF5) PICKED UP PAGE:
 DATASET ==>
```

#### (4) SORP 機能のメニュー画面

すると、SORP 機能の主要なサブコマンドを解説したメニュー画面が表示される。

```
----- SORP PRIMARY MENU -----
ENTER FOLLOWING SUBCOMMAND

· BEGIN - RETRIEVE TOP DATA SET
· NEXT - RETRIEVE NEXT DATA SET
· LOCATE - RETRIEVE SPECIFIED DATA SET
· DISPLAY - DIPLAY DATA SET INFORMATION
· HARDCOPY - OBTAIN OR RELEASE HARDCOPY UNIT
· KEEP - KEEP SPECIFIED DATA SET
· DELETE - DELETE SPECIFIED DATA SET
· HELP - SORP DESCRIPTION

USE END KEY (PF3) TO TERMINATE SORP
ROW SCROLL ==> 22 COLUMN SCROLL ==> 40
PAGE SIZE ==> 66 CODE ==> JEF

KEN50207I HARDCOPY FUNCTION IS NOT AVAILABLE
==> サブコマンド行
```

メニュー画面最下段の ==> で示されるサブコマンド行は次の画面に進んでも常に表示される。この SORP 機能では、この位置にサブコマンドを投入するか、あるいは PF キーを操作して SORP 機能を使用するわけである。

#### ○ バッチジョブの構成

バッチジョブは複数の出力用データセットから構成され、その出力用データセットは 1 ページまたは複数のページから構成されている。

その出力用のデータセットの内容は、FORTRAN77 のプログラムを実行した場合には

1. JOB LOG のデータセット  
処理の開始・終了時間・終了状態（正常終了かどうか）が表示されている
2. JCL STATEMENT LIST のデータセット  
JCL に関する情報が表示されている
3. SYSTEM MESSAGE LIST のデータセット  
詳しい処理状況が表示されている
4. プログラムの COMPILE 段階の情報のデータセット  
プログラムの COMPILE 状況が表示されている  
プログラムの文法エラーもこのデータセットに表示される
5. プログラムの LINK 段階の情報のデータセット
6. 計算結果のデータセット  
最も必要な処理結果が表示されている

の 6 つのデータセットからなる。なお、出力用データセット数はバッチジョブの種類によって異なり、通常は 3 - 6 個である。

#### (5) バッチジョブの画面表示と PF キー

[ PF 2 ]

SORP 機能では、これらのデータセットを次々表示させる場合に、PF 2 キーを用いる。これよりも先に出力用データセットが存在しない場合には、END OF DATA SET と表示される。

[ ENTER ] キー

ひとつの出力用データセットが複数のページからなる場合に、次のページを表示させる働きをする。これよりも先にページが存在しない場合には、END OF PAGE と表示される。

[ PF7・PF8 ] キー

前述のように出力用データセットの 1 ページは、66 行 132 カラムから構成されているので、端末の画面では一度に表示させることはできない。PF7 キーはそのページの上方を、PF8 キーはそのページの下方を表示させる機能をもつ。

[ PF10・PF11 ] キー

出力用データセットの 1 ページについて、PF10 キーはそのページの左方を、PF11 キーはそのページの右方を表示させる機能をもつ。



PF7・PF8・PF10・PF11 キーの働きは、FSO 画面の場合とほぼ同様である。

(6) 出力用データセットの表示

実際に PF2・ENTER・PF7・PF8・PF10・PF11 キーを用いて処理結果を表示させてみよう。

```
----- SORP PRIMARY MENU -----
ENTER FOLLOWING SUBCOMMAND
```

- ・ BEGIN - RETRIEVE TOP DATA SET
- ・ NEXT - RETRIEVE NEXT DATA SET
- ・ LOCATE - RETRIEVE SPECIFIED DATA SET
- ・ DISPLAY - DISPLAY DATA SET INFORMATION
- ・ HARDCOPY - OBTAIN OR RELEASE HARDCOPY UNIT
- ・ KEEP - KEEP SPECIFIED DATA SET
- ・ DELETE - DELETE SPECIFIED DATA SET
- ・ HELP - SORP DESCRIPTION

```
USE END KEY (PF3) TO TERMINATE SORP
ROW SCROLL ==> 22 COLUMN SCROLL ==> 40
PAGE SIZE ==> 66 CODE ==> JEF
```

```
KEN50207I HARDCOPY FUNCTION IS NOT AVAILABLE
=> サブコマンド行
```

この状態から、PF2 キーを押して最初の出力用データセットを表示させる。通常 JOB LOG が表示される。

(7) JOB LOG データセットの表示

```

JES JOB LOG -- SYSTEM E40B -- NODE N1

12.27.10 JOB 6621 *** F1234Z (JOB6621) 0000:START TIME=12.27.10
12.29.12 JOB 6621 CD=0000 *** F1234Z (JOB6621) 0000:END TIME=12.29.12
```

```
KEQ50901I END OF PAGE---*---3---*---4---*---5-NO 1 COLM 1 PAGE 1
=>
```

JOB LOG 情報の 3 行目に、終了状態 (CD= ) が表示されている。CD=0000 の場合は正常終了を示している。  
画面最下段から 1 行上に END OF PAGE と表示されているので、このデータセットには他のページは存在しない。したがって、ENTER キーを押してもこのページが表示されるだけである。  
また、このページの下方を表示させるには、PF8 キーを押す。  
終了状態を確かめた後、次のデータセットを表示させるために、PF2 キーを押す。

(8) JCL STATEMENTS LIST データセットの表示

```

E20 V10L20 <<< JCL STATEMENTS LIST >>> DATE 04/15/86
1 //F1234Z JOB CLASS=C, REGION=1024K
2 // EXEC FORT77, STEP=CLG
10 //FORT.SYSIN DD DSN=F1234.MEAN.FORT77, DISP=SHR
28 //GO.FT05P001 DD DSN=F1234.POP.DATA, DISP=SHR
```

```
KEQ50901I END OF PAGE---*---3---*---4---*---5-NO 2 COLM 1 PAGE 1
==>
```

JCL STATEMENTS LIST のデータセットを表示させた後、再び次のデータセットを表示させるために、PF2 キーを押す。

(9) SYSTEM MESSAGE LIST データセットの表示

```

<<< SYSTEM MESSAGE LIST >>>
```

```
26 JDJ686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP...
KDS70001I F1234 LAST ACCESS AT 11:37:21 ON 86.121
JDJ142I F1234Z FORT - STEP WAS EXECUTED - COND CODE 0000
JDJ373I STEP/FORT / START 86121.1227
JDJ374I STEP/FORT / STOP 86121.1227 CPU OMIN 00.18SEC SRB..
JDJ142I F1234Z LKED - STEP WAS EXECUTED - COND CODE 0000
JDJ373I STEP/LKED / START 86121.1227
JDJ374I STEP/LKED / STOP 86121.1227 CPU OMIN 00.64SEC SRB..
JDJ142I F1234Z GO - STEP WAS EXECUTED - COND CODE 0000
JDJ373I STEP/GO / START 86121.1227
JDJ374I STEP/GO / STOP 86121.1227 CPU OMIN 00.12SEC SRB..
JDJ375I JOB/F1234Z / START 86121.1227
JDJ376I JOB/F1234Z / STOP 86121.1227 CPU OMIN 00.95SEC SRB..
```

```
KEQ50901I END OF PAGE---*---3---*---4---*---5-NO 3 COLM 1 PAGE 1
==>
```

このデータセットには処理における COMPILE、LINK、実行の三段階が表示されている。さらに次のデータセットを表示させるために、PF2 キーを押す。

(10) プログラムの翻訳 (COMPILE) リスト

FORTRAN 77 V10L20 DATE 86.04.15 TIME 12.27.10

```
* CALCULATION OF MEAN
000001 DIMENSION SCORE(100)
000002 READ(5,1000) NUMBER
000003 1000 FORMAT(I10)
000004 READ(5,1010) (SCORE(I),I=1,NUMBER)
000005 1010 FORMAT(5F10.0)
000006 SUM=0.0
000007 DO 100 I=1,NUMBER
000008 SUM=SUM+SCORE(I)
000009 100 CONTINUE
000010 AMEAN=SUM/NUMBER
000011 WRITE(6,2000) AMEAN
000012 2000 FORMAT(2X,F10.2)
000013 STOP
000014 END
```

```
STATISTICS 14 STEPS, PROCEDURE SIZE= 122BYTES, PROGRAM NAME=MAIN
 15 LINES, PROGRAM SIZE= 989BYTE, DIAGNOSTIS = 0
 REMAINING SIZE= 908K BYTES,
-----1-----2-----3-----4-----5-NO 4 COLM 1 PAGE
==>
```

プログラムの COMPILE 状況のメッセージである。この例ではプログラムが短いので、プログラム全体が 1 ページで表示されてしまっている。プログラムが非常に長い場合や、プログラム以外にも、COMPILE にともなう大切な情報が出力されているので、このデータセットのメッセージには必ず全部目をとおすこと。

各ページの上方を表示させる場合には PF7 キー  
各ページの下方を表示させる場合には PF8 キー  
各ページの左方を表示させる場合には PF10 キー  
各ページの右方を表示させる場合には PF11 キー

では、このページ下方を PF8 キーを使って表示させよう。

```
.....
.....
.....
OPTIMIZE(2)
SIZE(992K)
XOPT(IL, AMOVE,NOMSG)
STATISTICS: 1 UNITS, 14 STEPS, 15 LINES,DIAGNOSTICS= 0,HIGHEST SEVER
END OF COMPILATION
-----1-----2-----3-----4-----5-NO 4 COLM 1 PAGE
==>
```

この画面では、END OF COMPILATION 表示の 1 行上に表示される

DIAGNOSTICS=  
HIGHEST SEVERITY CODE= (この数値は PF11 キーを押さないと表示されない)

の両方の数値が 0 でないと、プログラムに文法エラーが生じたことを示している。必ず確かめなければならない。

次のデータセットを表示させてみよう。

(1 1) LINK 段階のデータセット

```

LINKAGE EDITOR DATE 86.04.15 TIME 12.27.10

OPTIONS SPECIFIED - LET
SIZE1 <DEFAULT VALUE>= 262144
SIZE2 <DEFAULT VALUE>= 65536
MAX. LENGTH OF OUTPUT TEXT BLOCK = 18432
LINECOUNT = 60

.....
.....

-----*-----1-----*-----2-----*-----3-----*-----4-----*-----5-NO 5 COLM 1 PAGE
==>
```

LINK 段階のメッセージの後は、いよいよ処理結果を出力したデータセットである。同様に、PF2 キーを押す。

(1 2) 計算処理結果のデータセット

```

37093.30

-----*-----1-----*-----2-----*-----3-----*-----4-----*-----5-NO 6 COLM 1 PAGE
==>
```

これが最終の位置にある計算結果のデータセットである。画面上でよく結果を確認する必要がある。それには、前述のように PF7・PF8・PF10・PF11・ENTER キーを用いること。

出力結果がこの例のように 1 ページしかない場合は、PF2 キーを押すと

END OF DATA SET

と表示され、ENTER キーを押すと

END OF PAGE

と表示される

(1 2) 取りだし請求の実行

取りだし請求の実行には、画面最下段のサブコマンド行に

○ 用紙に出力させる場合には、END DSF [←]

○ 用紙に出力させず消去する場合には、END DEL [←]

とする。

END DEL と入力した場合には、SORP 機能は終了して READY 状態となる。END DSF と入力した場合には、SORP 機能の最終メニュー画面がでる。では、結果を用紙に出力させてみよう。

37093.30

KEN50901I END OF PAGE-----\*---3---\*---4---\*---5-NO 6 COLM 1 PAGE  
==> END DSF [←]

すると、SORP 機能の最終メニュー画面が表示される。

----- END END SUBCOMMAND MENU -----

| STATUS OF SYSOUT DATA SET(S)                    |       |       | OUTPUT WITH DEMAND SYSOUT FACILITY     |
|-------------------------------------------------|-------|-------|----------------------------------------|
| CHANGE/VERIFY DISPOSITION OF SYSOUT DATA SET(S) |       |       | DSF (YES OR NO)==> YES                 |
| DSNO                                            | CLASS | DISP  | ENTER CLASS NAME, IF YOU WANT TO       |
| 1                                               | D     | UNDEF | OUTPUT SYSOUT DATA SET(S) OF JOB WITH  |
| 2                                               | D     | UNDEF | DEMAND SYSOUT FACILITY                 |
| 3                                               | D     | UNDEF | CLASS NAME ==> D                       |
| 4                                               | D     | UNDEF |                                        |
| 5                                               | D     | UNDEF | ENTER DESTINATION NAME, IF YOU WANT TO |
| 6                                               | D     | UNDEF | CHANGE THE DESTINATION OF THIS JOB     |
|                                                 |       |       | DEST NAME ==>                          |

KEN50205I PRESS PF3 KEY TO TERMINATE DISPOSITON PROCESSING

この最終メニュー画面において、最も簡単な操作方法は PF3 キーを押して SORP 機能を終了させることである。この操作で、バッチジョブの全部の出力用データセットが1階オープン入出力室の NLP に出力され、計算機のスステイタスは SORP 機能が終了し READY 状態に戻る。

ところが、出力用データセットのうちではあまり必要でない情報もある。そのため出力用データセットのうちから実際に用紙に出力させるデータセットを選択できる。その場合には、画面の左側の DSNO に対応した DISP の UNDEF を KEEP に変更すればよい。

また、ジョブの出力先を指定する

DEMAND SYSOUT FACILITY  
CLASS NAME ==> D

において D となっているが、この D は1階オープン入出力室の NLP を指定している。この NLP が混雑している場合や、2階入出力機器室の磁気テープ装置を使用している場合には、この部屋の NLP を出力先に指定するとたいへん便利である。その際の指定方法は

DEMAND SYSOUT FACILITY  
CLASS NAME ==> A

と変更し、ENTER キーを押したのち最後に PF3 キーを押す。すると READY 状態に帰り SORP 機能によるバッチジョブの取りだし請求が終了する。ただし、2階の NLP が動作可能な状態になっているかを確認する必要がある。

実際に出力用データセットを 4・5 のみ残し、出力先も 2階 NLP に変更してみよう。

----- END END SUBCOMMAND MENU -----

STATUS OF SYSOUT DATA SET(S)

OUTPUT WITH DEMAND SYSOUT FACILITY

| CHANGE/VERIFY DISPOSITION OF SYSOUT DATA SET(S) |       |       | DSF (YES OR NO)==> YES                                                                                         |
|-------------------------------------------------|-------|-------|----------------------------------------------------------------------------------------------------------------|
| DSNO                                            | CLASS | DISP  | ENTER CLASS NAME, IF YOU WANT TO OUTPUT SYSOUT DATA SET(S) OF JOB WITH DEMAND SYSOUT FACILITY CLASS NAME ==> A |
| 1                                               | D     | UNDEF | ENTER DESTINATION NAME, IF YOU WANT TO CHANGE THE DESTINATION OF THIS JOB DEST NAME ==>                        |
| 2                                               | D     | UNDEF |                                                                                                                |
| 3                                               | D     | UNDEF |                                                                                                                |
| 4                                               | D     | KEEP  |                                                                                                                |
| 5                                               | D     | UNDEF |                                                                                                                |
| 6                                               | D     | KEEP  |                                                                                                                |

KEN50205I PRESS PF3 KEY TO TERMINATE DISPOSITON PROCESSING

上のようにメニュー画面上のパラメータを変更し、ENTER キーを押したのちPF3 キーを押す。  
READY

バッチジョブの NLP への出力には、NLP の込み具合によっては数分間かかる場合がある。ただし、NLP がダウンしている場合もあるので（例えば、NLP 用紙切れなど）、そのような場合にはすぐに情報処理センターの職員へ申し出ること。

## 7 データセットの管理

### 7-1 データセットの編集に関わる EDIT 以外のコマンド

#### (1) MERGE コマンド

これは、あるデータセットの内容を別のデータセットの中へ組み込むためのコマンドである。例えば、複数のデータを個別のデータセットに作成していたが、一つのデータセットにまとめたいという時などに用いる。ただし、NONUM のデータセットを MERGE させる時には、必ずオペランド NONUM を付け加える。

データセット DATA1.DATA の内容を DATA.DATA に組み込む場合

```
READY
MERGE DATA1.DATA DATA.DATA NONUM [←]
READY
```

#### (2) DELETE コマンド

このコマンドはデータセットを消去するコマンドであるからくれぐれも慎重に使用すること。いらなくなったデータセットは、このコマンドを用いて速やかに消去することが望ましい。もちろん、不要であることや消去するデータセット名を確かめたうえでのことである。こうした注意を怠った場合、どうなるかは各自でよく考えるべし。内容識別修飾子を省略した場合、内容識別修飾子の異なる同一データセット名が複数ある時には、DELETE コマンドの実行が保留される。

#### ○ 順データセットの場合

```
READY
DELETE JAPAN.DATA [←]
ENTRY (A) F1234.JAPAN.DATA DELETED
```

#### ○ 区分データセットの場合

区分データセットを消去する場合には、代表名（データセット名）のみを指定すると一括して全てのメンバーも消去されてしまうので特に注意を要する。代表名（データセット名）AMERICA・メンバー名 CITY という区分データセットを消去する場合には

```
READY
DEL AMERICA.DATA(CITY) [←]
ENTRY (A) F1234.AMERICA.DATA(CITY) DELETED
```

とする。

なお、上の例のように DELETE コマンドは DEL という省略形で使用してもよい。

#### (3) COPY コマンド

データセットに関わる処理作業の際のバックアップのためや、新しいデータセットの作成に利用すると便利である。このコマンドの基本形は

```
COPY NA1.内容識別修飾子 NA2.内容識別修飾子 [←]
 NA1:既成のデータセット名
 NA2:新データセット名
```

である。複写先の新しいデータセットの器を EDIT コマンド・ALLOC コマンドなどによって予め作成しておく必要はない。

```
READY
COPY 'F1246.JAPAN.DATA' JAPAN.DATA [←]
```

上の例は、他の課題番号のデータセット（必ず、両端を「'」で囲む）を複写した例である。この場合にはあらかじめ他の課題番号のデータセットの読みとり許可を得ておかなければならない（7-3 「データセットの保護（読み取り・書込み許可）」）。

○ 区分データセットの場合

区分データセット AMERICA.DATA を全メンバーともそっくり区分データセット USA.DATA に複写してみよう。この場合には、代表名（データセット名）のみを指定する。

```
READY
COPY AMERICA.DATA USA.DATA [←]
```

また次のように、区分データセットを順データセットにも複写できる。ただし、同一代表名の区分データセットのもとで、あるメンバーを別のメンバーとして直接複写することは出来ない。

```
READY
COPY AMERICA.DATA(CITY) AMRCITY.DATA [←]
```

(4) RENAME コマンド

RENAME コマンドはデータセット名を付けかえるコマンドである。次に、データセット名を JAPAN.DATA から NIHON.DATA へ変更する例を示す。

```
READY
RENAME JAPAN.DATA NIHON.DATA [←]
```

○ 区分データセットの場合

区分データセットの場合には、代表名（データセット名）やメンバー名の変更ができる。次に示すものが代表名（データセット名）の変更である。

```
READY
RENAME AMERICA.DATA USA.DATA [←]
```

メンバー名の変更の場合には

```
READY
RENAME AMERICA.DATA(CITY) AMERICA.DATA(METRO) [←]
```

とする。

7-2 既存のデータセットの一覧を出力させるコマンド  
FLIST コマンド・MLIST コマンドを中心に

(1) LISTC 及び DSL コマンド

どちらもほぼ同じような働きをするコマンドである。ただし、DSL コマンドのほうが情報量が多い。LISTC コマンドでは、自分の課題番号に属しているデータセット名のみが表示される。PS・PO 形式の別、レコード形式などは表示されない。

他方の DSL コマンドでは、データセット名以外にもそれぞれの属性（順・区分データセットの別、レコード長、ブロック長、スペース量、作成日時など）を表示させることができる。

```
READY
LISTC [←]
IN CATALOG:SYS1.EDF.MCAT
F1234.AMERICA.DATA
F1234.CARD.CNTL
F1234.COMMAND.CLIST
F1234.MEAN.FORT77
F1234.JAPAN.DATA
```

```
READY
DSL [←]
```



| DATASET NAME        | P.SPACE  | L.SPACE | VOLUME | CREATE | EX | ORG | RECFM | BLKSIZE | LRECL |
|---------------------|----------|---------|--------|--------|----|-----|-------|---------|-------|
| F1234.AMERICA.DATA  | 567      | 135     | PUB013 | 851224 | 1  | PO  | FB    | 3120    | 80    |
| F1234.CARD.CNTL     | 27       | 27      | PUB011 | 840122 | 2  | PO  | FB    | 3120    | 80    |
| F1234.COMMAND.CLIST | 27       | 27      | PUB012 | 860417 | 3  | PO  | FB    | 3120    | 80    |
| F1234.MEAN.FORT77   | 54       | 27      | PUB013 | 830708 | 1  | PS  | FB    | 3120    | 80    |
| F1234.JAPAN.DATA    | 135      | 135     | PUB011 | 850630 | 1  | PS  | FB    | 3120    | 80    |
| -----               |          |         |        |        |    |     |       |         |       |
| TOTAL               | 5        | 810     | 351    |        |    |     |       |         |       |
|                     | DATASETS | KB      | KB     |        |    |     |       |         |       |

データセットの保持に対する情報処理センターの課金は、磁気ディスク装置の占有量に当る P.SPACE 量（この場合は 810KB）について、100KB ごとに 1 月 50 円かかる。しかし、実際にデータセットの保持のため使用しているスペースは L.SPACE 量（この場合は 351KB）に当る。

磁気ディスク装置の有効利用の観点からも使用料金の観点からも、実際に使用していない領域は解放しておかなければならない。それには、@RELEASE コマンドを用いる。ただし、このコマンドは計算機に大きな負担をかけるので、できるだけセッション開始時に行なうとよい。

```

READY
@RELEASE [←]
F1234.AMERICA.DATA RELEASED
KEQ52640I CONDENSE PROCEEDING
.....
TOTAL RELEASE SPACE = 459KB
READY

```

## (2) FLIST コマンド

FLIST コマンドはデータセットの一覧を表示させるとともに、メニュー一覧に表示されたデータセットに対して、コマンドを入力して種々の処理を行なうことができる。

```

READY
FLIST [←]

```

```

==>
-- FLIST MENU -- V20/L04 -- PREFIX(F1234) - 86/09/09 - ID(F1) -- 1/ 18
NAME TYPE ORG INPUT FIELD RF LRECL BLKSIZE EX SPACE % CREATE VOLUME
AMERICA DATA PO F 80 3120 1 567 24 851224 PUB013
CARD CNTL PO F 80 3120 2 27 ** 840122 PUB012
COMMAND CLIST PO V 255 3120 3 27 ** 860417 PUB012
MEAN FORT77 PS F 80 3120 1 54 50 830708 PUB013
JAPAN DATA PS F 80 3120 1 135 ** 850630 PUB011
** END **

```

DISP=>PF6, FRWD=>PF8, BACK=>PF7, REPF=>PF5, , SORT=>PFR, SWAP=>PF2, TERM=>PF3

これが FLIST コマンドによるメニュー一覧である。メニュー画面最上段 ==>（完全入力フィールドと呼ばれている）またはメニュー項目の右側の INPUT フィールド（簡略入力フィールドと呼ばれている）に、TSS コマンドまたは FLIST 支援システムコマンドを投入・実行することができる。

では、MEAN.FORT77 のデータセットを呼びだし、編集してみよう。まず、カーソルを当該のデータセット右側の入力フィールドへ移動させ、EDIT とタイプする。そして、ENTER キーを押す。

==>

```

-- FLIST MENU -- V20/L04 -- PREFIX(F1234) - 86/09/09 - ID(F1) -- 1/ 18
NAME TYPE ORG INPUT FIELD RF LRECL BLKSIZE EX SPACE % CREATE VOLUME
AMERICA DATA PO F 80 3120 1 567 24 851224 PUB013
CARD CNTL PO F 80 3120 2 27 ** 840122 PUB012
COMMAND CLIST PO V 255 3120 3 27 ** 860417 PUB012
MEAN FORT77 PS EDIT [←] F 80 3120 1 54 50 830708 PUB013
JAPAN DATA PS F 80 3120 1 135 ** 850630 PUB011
** END **

```

DISP=>PF6, FRWD=>PF8, BACK=>PF7, REPF=>PF5, , SORT=>PFR, SWAP=>PF2, TERM=>PF3

すると、4-2 「EDIT-FSO によるデータセットの修正・追加方法」に示したものと全く同じ MEAN.FORT77 の EDIT-FSO の画面が表示される。以後全く同様に EDIT-FSO によるデータセットの編集を行なうことができる。

データセットの編集が終了し、SAVE サブコマンドおよび END サブコマンドを投入すると、再び FLIST メニュー画面に戻るので、PF3 キーを押すと FLIST 支援システムは終了して、READY 状態に戻る。

この FLIST コマンドによるデータセットの編集はとても便利で、初心者には面倒なオペランド ( FIXED など) を入力する必要がない。

### (3) MLIST コマンド

これも、LISTCコマンドでは表示されない区分データセットのメンバー名を表示させ、編集・実行などをさせる非常に便利なユーティリティコマンドである。

READY

MLIST CARD [←]

==>

```

-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> -
COMPPS ==> -
PRINTPS ==> -
SUBCHK ==> -
** END **

```

上のような区分データセットの処理のためのメニュー画面がでるので、EDIT コマンド・SUBMIT コマンドなどをタイプして、コマンドを実行することができる。

```

-- MLIST -- V20/L04 -- F1234.CARD.CNTL ----- ID(M1) -- 1/ 11
BATCH ==> EDIT [←] -
COMPPS ==> -
PRINTPS ==> -
SUBCHK ==> -
** END **

```

ENTER キーを押す。すると下のような画面が表示されるので、

```

KEQ23381 DATA SET CARD.CNTL(BATCH) NOT LINE NUMBERED, NONUM ASSUMED

```

もう一度 ENTER キーを押すと、EDIT-FSO 画面が表示される。以下はEDIT-FSO操作と全く同様に行なう。

MLIST メニューを終了するには、PF3 キーを押すと READY 状態にもどる。

### 7-3 書込み・読みとり許可（データセットの保護）

センターでは、利用者のデータセットは RACF ( RESOURCE ACCESS CONTROL FACILITY ) によって管理されている。これはセンターの RESOURCE をすべての利用者が平等かつ有効に活用でき、他の利用者から故意あるいは誤ってデータセットが破壊されることを防ぎ、さらに機密保護強化のためのものである。

したがって、他の課題番号のデータセットは、その持主がその課題番号のセッションのもとで許可を与えない限り、他の課題（利用者が同一であっても）からはアクセスできない。また、このアクセス権には特定アクセス権と公衆アクセス権の 2 種類がある。

MT (磁気テープ) 上のデータセットをホスト上の自分の課題のデータセットに書き込んだり、ホスト上の自分の課題のデータセットを MT に書き込んだりする場合には、自分の課題番号の管理下で行なうのでアクセス権の許可は不要である。

以下に、JAPAN.DATA というデータセットのアクセス権の許可に関する方法を述べる。

#### (1) 読みとり許可を与える場合

これは、自分の課題番号のプログラム・データ・JCL などのデータセットを他人に複写させる場合などに用いる。

#### ○ 特定の他の課題に対して許可を与える場合（特定アクセス権）

```
READY
PERMIT JAPAN.DATA ID(F1246) ACCESS(READ) [←]
```

この JAPAN.DATA は課題番号 F1246 の利用者に対してのみ読み取り許可を与えたことになる。

#### ○ 不特定多数の利用者に対して許可を与える場合（公衆アクセス権）

この場合は、くれぐれも注意が必要である。しかし、ホスト上のデータセットを読んで OCR 端末のフロッピーディスクへ送信する場合には、自分の課題のデータセットであっても公衆アクセス権が必要で、予め下記の作業が必要である。

```
READY
ALTDSD JAPAN.DATA UACC(READ) [←]
```

これで、JAPAN.DATA は他のすべての課題番号からの（第 3 者からも）COPY コマンドなどによるアクセスが可能になった。

#### (2) 書込み許可を与える場合

書込み許可は、むやみに他の課題に対して与えると自分のデータセットに重大な支障をきたすことがあるので、くれぐれも慎重にすることが必要である。

#### ○ すべての他の課題に対して許可を与える場合（公衆アクセス権）

これは、OCR (光学読み取り) 装置の端末を利用してフロッピーディスクに保存されているデータセットをホスト上の自分の課題のデータセットとしてへ送信する場合に用いる。

```
READY
ALTDSD JAPAN.DATA UACC(UPDATE) [←]
```

これで、JAPAN.DATA へ OCR (光学読み取り) 装置からの送信が可能になった。

## 8 困った時の対処方法

計算機を使用している最中に思わぬ事態に直面することがあるが、どういう場合であってもあせらずに落ち着いて行動しなければならない。慣れないうちに次のようなマークが出てくると慌ててしまいがちである。

### (1) 何らかのキーを押しても、カーソルが移動しない場合

画面最下行の左の表示  
X ←?→ =)

RESET キーを押した後、カーソルを右へ移動させてみる。このマークが出た時は、入力できない位置に文字を入力しようとしたと考えられる。

X SYSTEM =)

RESET キーとともに、ALT キーと DUP キーを3個同時に押してみる。このマークは、画面と画面の変わり目に瞬間的に表示されることがよくあるが、長く表示されるような時に上の方法を用いるとよい。

### (2) 無限ループに入ってしまった場合

\*\*\*  
X SYSTEM =)

RESET キーを押した後、ALT キーと DUP キーを同時に押してみる。

### (3) REENTER+ というメッセージが出た場合

どのオペランドの指定を間違えて REENTER を促されているかわかりにくいので、ALT キーと DUP キーを同時に押してアテンション割りこみを行って、一度 READY 状態に戻しもう一度最初からコマンドの入力を行なう方が効率的である。