

ISSN 0452-2982

UDC 681.3.02

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-444

次期計算機システム用ジョブ制御マクロの設計

土 屋 雅 子 ・ 末 松 和 代 ・ 畑 山 茂 樹

1981年9月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

次期計算機システム用ジョブ制御マクロの設計*

土 屋 雅 子** 末 松 和 代** 畑 山 茂 樹**

1. はじめに

計算機システムにジョブを投入する手続きは、各システム独自のジョブ制御言語 (JCL: Job Control Language) の文法に則ったものでなければならない。JCLは各メーカーによって異なるだけでなく、同一メーカーでもオペレーティングシステム (OS) 毎に異なる。しかも、JCLにはFORTRAN言語のような互換性もなく、ユーザに直接ハードウェアを意識させるという負担を強いている。計算機システムのこの硬直性は大いに批判され、改善されなければならないが、現状は上記の通りである。¹⁾

このため、計算機システムを移行した場合、ユーザは新たなジョブ投入手続きを習得しなければならない。しかし、全ユーザに対して、JCLの文法に則り、新たなジョブ投入手続きの記述を強いるのは余りにも負担が過大すぎる。

しかし幸いにして、OSにはジョブ制御マクロ機能というものが用意されている。これはあらかじめ定義しておいたジョブ制御文の一連の記述体 (以下、マクロと略称) を簡単なジョブ制御文によって呼び出すことを可能にする機能である。

この機能を活用し、ほとんどのユーザには十分といえるだけのマクロを計算センタで用意しておけば、ユーザは新たなJCLを意識することなく、ジョブ処理に必要なマクロを呼び出すための数枚のジョブ制御文を用意するだけで、新たな計算機システムにジョブを投入することができる。ほとんどのユーザは各マクロの設計仕様と、そのマクロパラメータの記述形式とを習得するだけでよい。従って、JCLの移行のためユーザに強いる負担は大巾に軽減できるであろう。

本稿は、現有の航技研計算機システム (FACOM 230-75 APU, MONITORⅦシステム²⁾) が移行し

た場合、計算センタではどのようなマクロを次期システム用に用意すべきか、またどのようなマクロパラメータ記述形式をとればユーザが利用しやすいか、さらにこれらのマクロを使用したときジョブ投入手続きはどうなるか、などを2つのOS (OSⅣ / X8³⁾ と F4⁴⁾) を例にとり、検討したものである。

2. 用 語

MONITORⅦシステムとOSⅣシステムとで異なる主な用語は次の通りである。以後の議論は各システムに固有な言葉で行うことにする。

MONITORⅦ	OSⅣ
ジョブ制御マクロ	ジョブ制御マクロ (X8)
	カタログドプロシジャ (F4)
相対形式プログラム	オブジェクトモジュール
実行形式プログラム	ロードモジュール
分割型順編成	区分編成
ファイル	ファイル (X8)
	データセット (F4)
エレメント	メンバ
FD文	FD文 (X8)
	DD文 (F4)
FTMAIN	MAIN
ファイル定義名	アクセス名 (X8)
	dd名 (F4)

3. オブジェクトモジュールとロードモジュール

マクロ機能一般の検討を始める前に、重要なマクロ機能の一つである個人ファイルの作成法並びに参照法に関する仕様を決定する上で必要となる説明をまず行っておく。

MONITORⅦにおける相対形式プログラム、実

* 昭和56年7月31日受付

** 計算センター

行形式プログラムはOSⅣではそれぞれオブジェクトモジュール、ロードモジュールと呼ばれる。即ち、オブジェクトモジュールはコンパイラの出力であり、^{5),6)} ロードモジュールはリンケージエディタの出力である。^{7),8)}

ユーザのプログラムはソースプログラムの状態で保存されるだけでなく、オブジェクトモジュールまたはロードモジュールの状態でも保存される。ところが、MONITORⅦとOSⅣとではオブジェクトモジュールとロードモジュールの作成・参照法が以下の如く異なる。

3.1 MONITORⅦの場合

3.1.1 相対形式プログラム

FORTRAN H コンパイラは相対形式プログラムを、自動的に翻訳単位毎に分割し、主プログラムのエレメント名にはFTMAINを、副プログラムのエレメント名には副プログラム名を付加して、分割型順編成ファイルとして作成する。従って、相対形式プログラムを保存する場合には保存先のファイル名のみを指定し、リンケージエディタの入力とする場合にはファイル名とエレメント名（即ち、プログラム名）を指定する（ファイル名のみを指定すると、その全エレメントがリンケージエディタの入力となる）。

また、リンケージエディタの自動呼出しライブラリを作成するには、LIBE制御文の“EDIT, SL”を用いて、相対形式プログラムを再編集しなければならない。

3.1.2 実行形式プログラム

リンケージエディタは複数個の相対形式プログラムを入力とし、1個の実行形式プログラムを、分割型順編成ファイルとして作成する。このため、実行形式プログラムを保存または参照する場合にはファイル名とエレメント名を指定する。

なお、リンケージエディタには実行形式プログラムを入力することはできない。

3.1.3 まとめ

以上のように、MONITORⅦシステムの相対形式プログラムと実行形式プログラムの作成・参照法

はユーザにとって融通性のないものであるといえる。

3.2 OSⅣの場合

3.2.1 オブジェクトモジュール

FORTRAN 77 コンパイラは第3.1.1項の如く翻訳単位毎に分割せず、入力ソースプログラムの翻訳結果全体を1つのオブジェクトモジュールとして、順（または区分）編成で作成する（区分編成で作成する場合はメンバ名の指定も必要である）。

このため、MONITORⅦシステムと異なり、OSⅣシステムではオブジェクトモジュールの保存は行わず、ユーザライブラリは全て次に述べるロードモジュールの状態で保存するよう、ユーザを指導することが必要である。こうすれば、オブジェクトモジュールはシステム上、一時的順編成ファイル（データセット）のみとなり、入出力処理効率の向上を計ることができる。

なお、やむを得ずオブジェクトモジュールの保存・参照を行う場合には、次の点に十分留意しなければならない。

(i) 保存先のファイル名のみ指定した場合、オブジェクトモジュールは順編成で作成され、メンバ名も指定した場合には区分編成で作成される。

(ii) オブジェクトモジュールはそれを作成した時のプログラム全体がリンケージエディタの入力となり、1個のソースプログラム毎に作成したオブジェクトモジュール以外、MONITORⅦシステムのようにプログラム単位での入力とすることができない。

(iii) OSⅣシステムのシステムライブラリ^註はロードモジュールの集合とすべきであり、またオブジェクトモジュールとロードモジュールとではファイル（データセット）属性が異っている。このため、オブジェクトモジュールは自動呼出しライブラリとして使用できない。

3.2.2 ロードモジュール

MONITORⅦシステムと異なり、リンケージエディタにはオブジェクトモジュールのみでなくロードモジュールも入力することができる。その出力は区分編成で管理される。この機能を活用し、OSⅣ

註) システムライブラリとはFORTRANライブラリ、SSL, PSP等の集合をいう。

システムでは実行形式プログラムだけでなくユーザライブラリもロードモジュールの状態と保存することに統一し、ユーザがオブジェクトモジュールを意識する必要のないものとすべきである。以下で詳述するように、ロードモジュール状態のユーザライブラリには、MONITORⅦシステムの相対形式プログラムが持つ全ての機能を包含させることができる。

また、MONITORⅦシステムと異なり、どのような目的のロードモジュールを作成し、どのようにそれを参照するかは、ユーザの意志に委ねられている。即ち、FORTRAN 77 コンパイラのオプションでNAMEを指示するか否か、リンケージエディタのオプションでNOCALL (X8の場合)、NCAL(F4の場合)、LETを指示するか否かの選択がユーザに委ねられている。ここで、NAMEオプションは各プログラム単位のオブジェクトモジュールの最後に、リンケージエディタのNAME制御文を出力する旨の指示である(図1)。NOCALL(NCAL)オプションは未解決な外部参照名に対するライブラリの

自動呼出し機能を抑制する旨の指示である。^(註) LETオプションは未解決な外部参照名など、重大度2のエラーを検出してもリンケージエディタのエラーとせず、実行可能とする旨の指示である。なお、NOCALL(NCAL)の指示があると、LETの指示は無意味となる。

このように、上記3つのオプションを自由に選択できることと、リンケージエディタにロードモジュールを入力できることから、非常に多種類のロードモジュールが作成可能であるが、この融通性は逆にユーザを徒に混乱させる。そこで、ユーザにとって必要かつ十分と考えられるロードモジュール作成・参照方法を以下に詳述する。ユーザは次のいずれかを選択すれば十分であろう。

(A) 実行形式プログラムの作成法

実行形式プログラムとしてロードモジュールを保存したい場合、使用目的に応じて次の2つの作成法

(註) このオプションが指示されると、未解決の外部参照名があっても実行可能とされる。

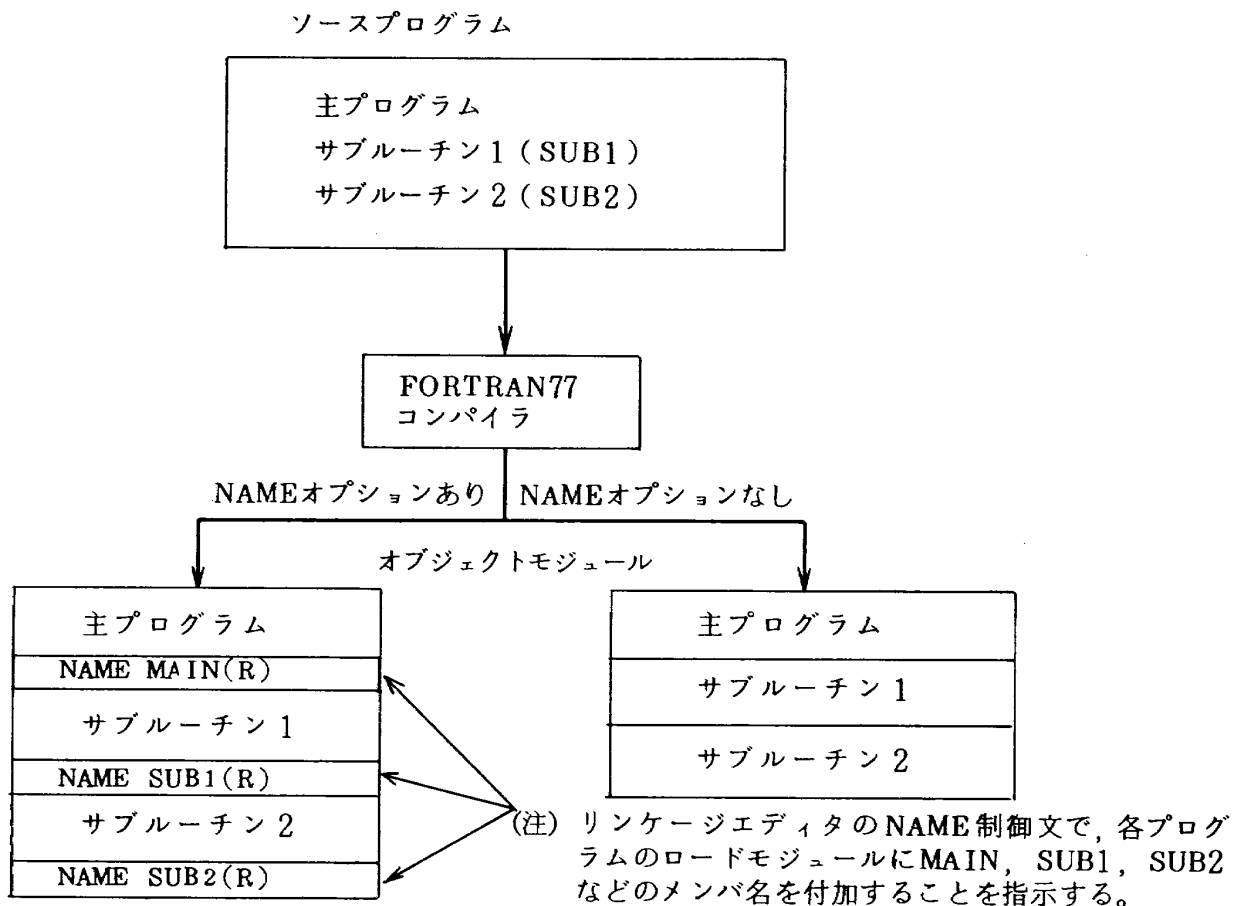


図1 FORTRAN 77 コンパイラのNAMEオプション

がある。

(A.1) 作成法 1

上記オプションを一切指示せず、ファイル(データセット)名とメンバ名を指定してロードモジュールを作成する。こうして作成されたものをLOAD 1と呼び、必ず主プログラムを含むものとする。LOAD 1はMONITOR VII システムの実行形式プログラムに相当するが、第D項で述べるように、それ以上の機能を有する。図2の如く、LOAD 1はプログラムが単純構造であるならば、全ての外部参照が解決されている。

(A.2) 作成法 2

LETオプションのみを指示し、ファイル(データセット)名とメンバ名を指定してロードモジュールを作成する。こうして作成されたものをLOAD 2

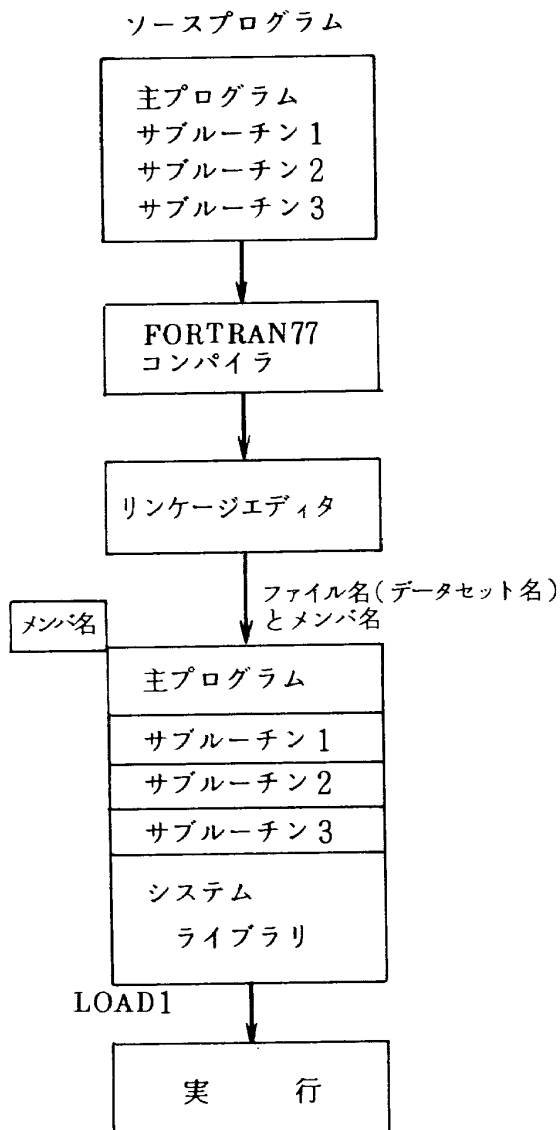


図2 ロードモジュール作成法 1

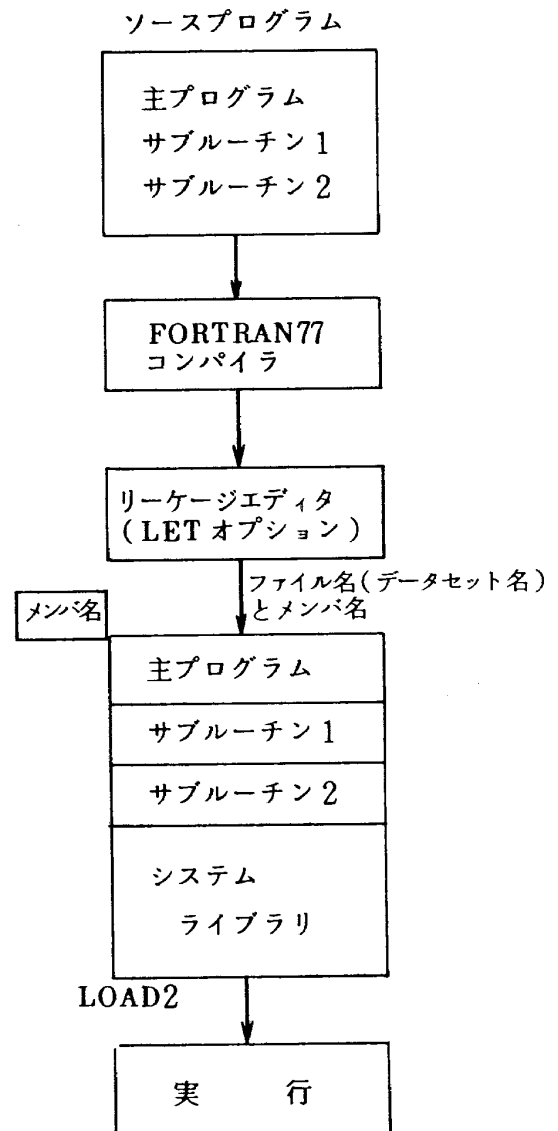
と呼び、必ず主プログラムを含むものとする(図3)。LOAD 2は未定義な外部参照名があっても、プログラムを実行したいという場合に有用となる。例えば、入力データに依存して主プログラムの実行の流れが変わり、一部のサブルーチンが呼ばれることのないような場合、それら不必要なサブルーチンを取りはずした小さな実行形式プログラムが作成できる。

(B) ユーザライブラリの作成法

ユーザライブラリとしてロードモジュールを保存したい場合、使用目的に応じて次の2つの作成法がある。

(B.1) 作成法 3

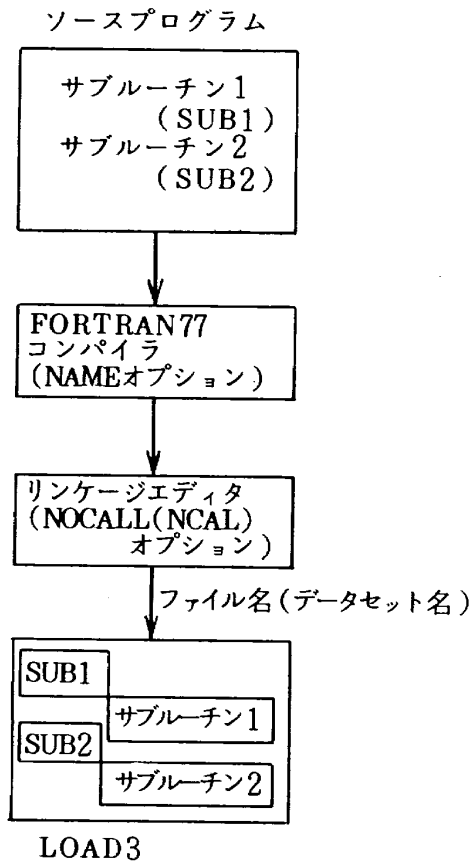
NAMEとNOCALL(NCAL)オプションを指示



(注) 主プログラムが呼び出すサブルーチン1と2以外は未定義とする。

図3 ロードモジュール作成法 2

し、ファイル（データセット）名のみを指定してロードモジュールを作成する。こうして作成されたものをLOAD 3と呼び、一般には主プログラムを含まないものとする。図4に示すように、LOAD 3は自動的にプログラム単位毎に分割され、主プログラムのメンバ名にはMAINを、副プログラムのメンバ名



(注) これらのメンバ名 (SUB1, SUB2) はシステムが付ける。

図4 ロードモジュール作成法3

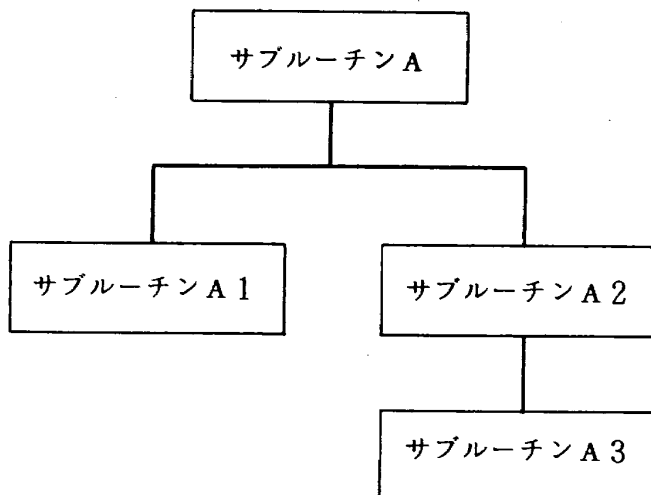


図5 例

には副プログラム名を付加して作成される。このように、LOAD 3はMONITOR VII システムの相対形式プログラムと同様な取り扱いができるだけでなく、第C項で述べるように、それ以上の機能を有している。

(B.2) 作成法4

NOCALL(NCAL) オプションのみを指示し、ファイル（データセット）名とメンバ名を指定してロードモジュールを作成する。こうして作成されたものをLOAD 4と呼び、一般には主プログラムを含まないものとする。

LOAD 4は図5の例のように、サブルーチンAと、そのみが呼び出すサブルーチンA1, A2, A3をまとめて一つのメンバ名で管理したい場合に有用である（図6）。なお、メンバ名はサブルーチンAのプログラム名としなければならない。

(C) ユーザライブラリの参照法

ユーザライブラリであるLOAD 3と4をリンケージエディタの入力とする場合、使用目的に応じて次の3つの参照法がある。

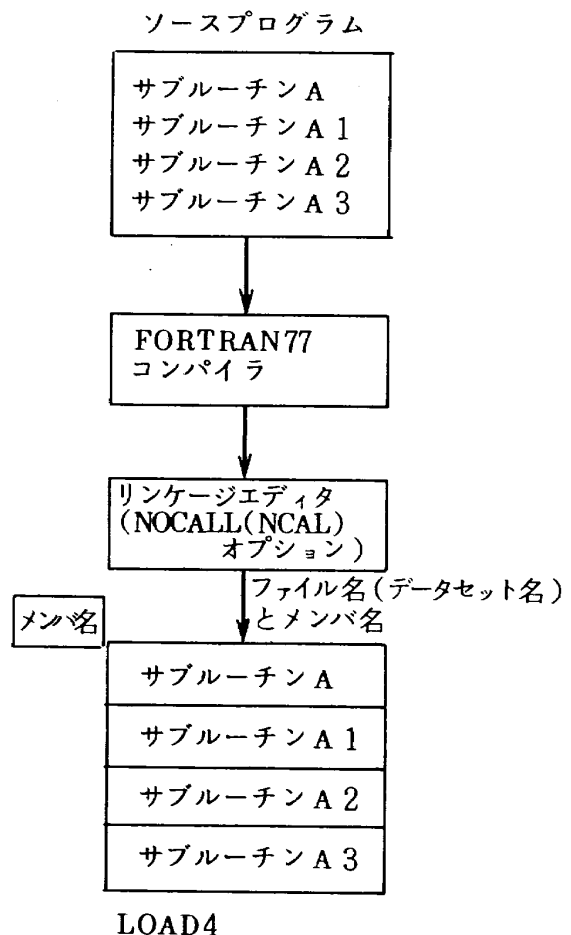


図6 ロードモジュール作成法4

(C.1) 参照法 1

MONITORⅦシステムと異なり、LOAD 3と4はそのまま自動呼出しライブラリとして使用できる。自動呼出しライブラリとして使用する場合、ファイル(データセット)名のみを指定してリンケー

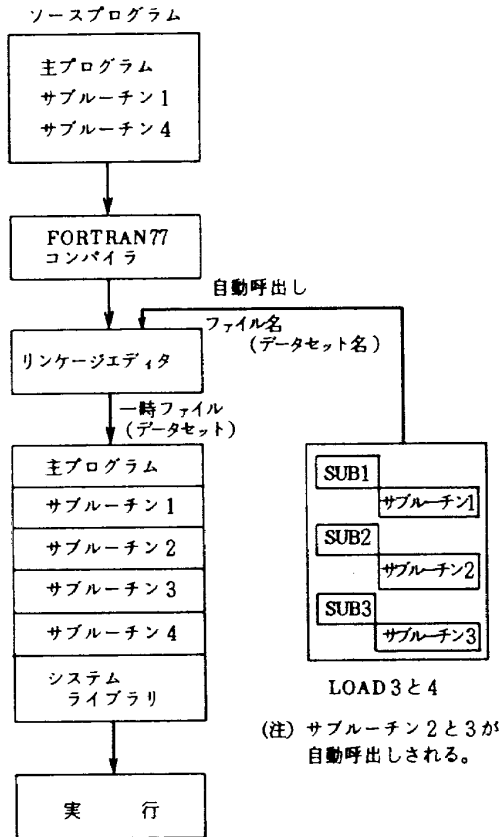


図7 ロードモジュール参照法1

ジェディタの入力とすればよい(図7)。

(C.2) 参照法 2

LOAD 3と4はファイル(データセット)名とメンバ名を指定してリンケージェディタの入力とすれば、指定された全てのプログラムが有効となる(図8)。ただし、同一メンバ名があると、先にリンケ

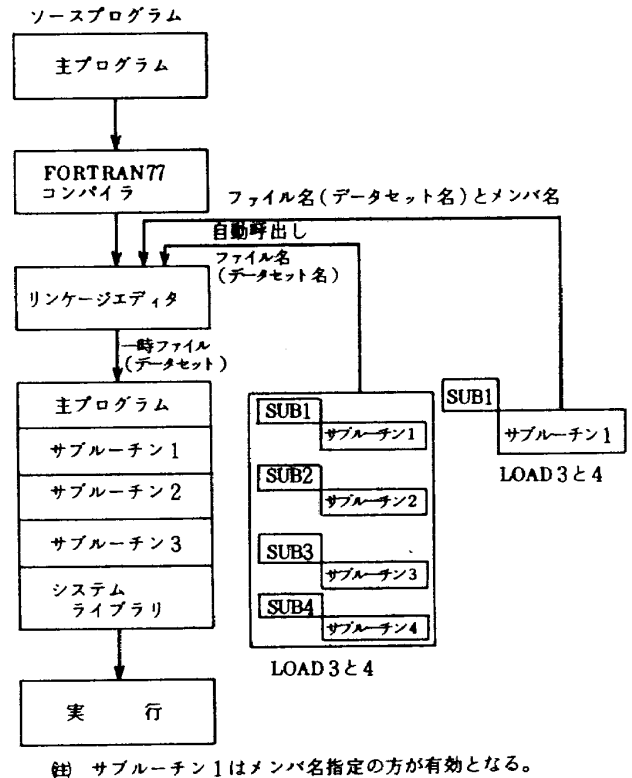
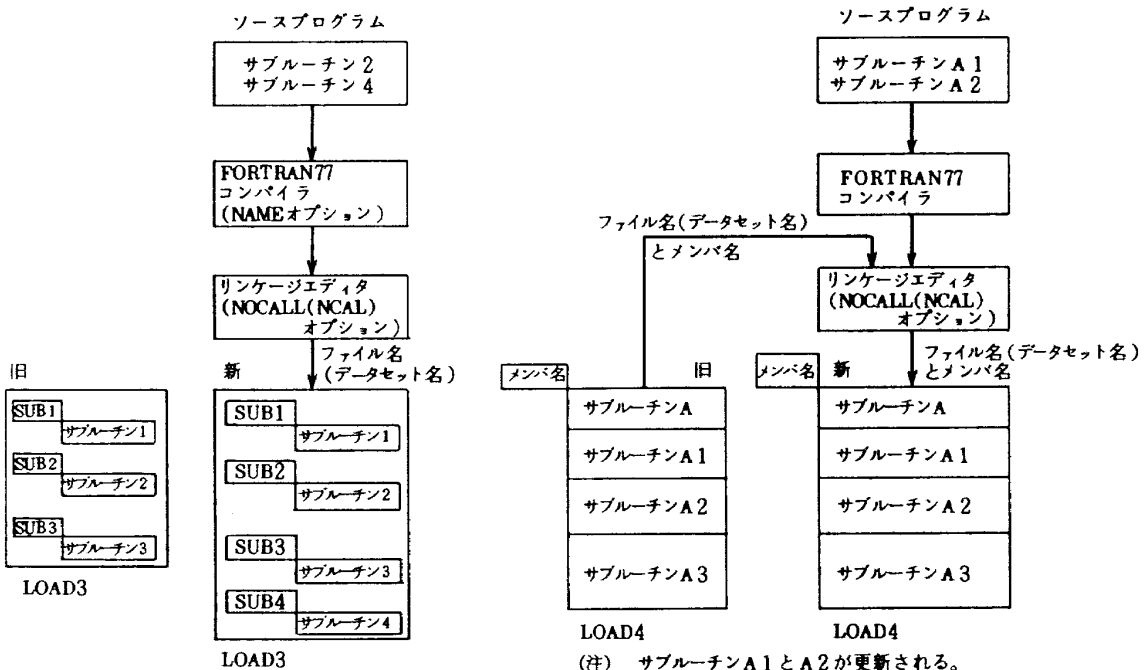


図8 ロードモジュール参照法2



(注) SUB2は更新され、SUB4が追加される。

(i) LOAD3の場合

(注) サブルーチンA1とA2が更新される。

(ii) LOAD4の場合

図9 ロードモジュール参照法3

ージエディタに入力されたプログラムのみが有効となる。

(C.3) 参照法3

LOAD 3の既存メンバの入れ替えや新メンバの追加、LOAD 4の一部プログラムの更新は図9のようにして行うことができる。更新したいプログラムを先にリンケージエディタの入力とすればよい。

(C.4) 注 意

LOAD 3と4を実行してはならない。実行すると直ちにエラーとなる。

(D) 実行形式プログラムの参照法

実行形式プログラムであるLOAD 1と2をリンケージエディタの入力とする使用目的に、次のような参照法がある。

(D.1) 参照法4

図10のように、LOAD 1と2の一部サブルーチンを新しいものと置き換えることができる。更新し

たいプログラムを先にリンケージエディタへ入力する。

(D.2) 注 意

LOAD 1と2を自動呼出しライブラリとして使用してはならない。使用した場合にはリンケージエディタで未定義記号のエラーとなる。

3.2.3 まとめ

以上のように、ロードモジュールはMONITOR VIIの実行形式プログラムにくらべて、利用価値が増大しており、ユーザにとって融通性のある使い方が出来る。しかし、ロードモジュールの作成法・参照法としては上記の4つを知っていれば十分であろう。

なお、作成法1は上記オプションを一切意識する必要がなく、故にMONITOR VIIとの違いを感じない。また、作成法3はMONITOR VIIシステムと著しい違いがあるが、計算センタでLOAD 3を作成するマクロを用意し、ユーザがこの違いを意識する必要のないものとするべきである。また、LOAD 2と4は特殊用途であるから、ユーザ自らが意識して作成すべきである。

4. マクロの設計仕様

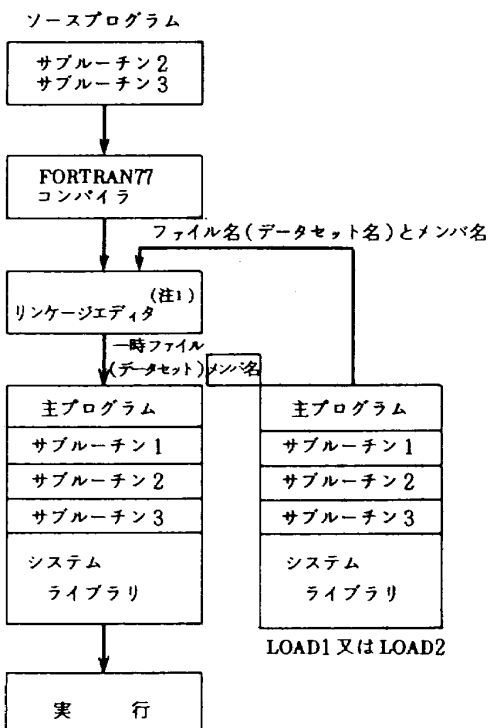
計算センタでは、次期システム用マクロとして以下のものを用意する必要がある。また、以下のマクロがあれば、ほとんどのユーザのジョブ投入手続のためには十分と思われる。

4.1 ジョブ識別用マクロ

ジョブの先頭を示す文と、ジョブの最後を示す文が必要である。なお、OSによってはジョブ識別用制御文をマクロ化できないものがある。下記のマクロ名はマクロ化できる場合の名称である。

4.1.1 KJOB文

ジョブの先頭を示す文であり、ユーザのシステム使用権、個人ファイル作成・消去権確認のためのユーザ登録名、ユーザがジョブに与える各種要求(ジョブクラス、ジョブ打切時間、ジョブ出力先など)を記述できるものとする。^(注)



(注1) LOAD1のときはLETオプションなし。
LOAD2のときはLETオプションあり。

(注2) サブルーチン2と3が更新される。

図10 ロードモジュール参照法4

(注) MONITOR VIIシステムのNO文とKJOB文とを、一枚のKJOB文に集約する。なお、次期システムは仮想記憶方式となるから、“コア占有語数”パラメータは不必要となる。ユーザはジョブクラス毎に設定される仮想記憶の使用可能容量のみを意識して、プログラムを作成すればよい。

4.1.2 JEND文

ジョブの最後を示す文である。

4.2 プログラムの翻訳・結合編集・実行用マクロ

プログラムの翻訳・結合編集・実行を指示するマクロの他に、ファイル（データセット）の作成・参照を指示するマクロが必要である。また第3.2.3項で詳述したように、ユーザライブラリを簡単に作成できるマクロの用意も必要である。

4.2.1 FORTC文

FORTRANソースプログラムの翻訳を指示する文であり、入力ソースプログラムが複数個指定可能とし、リスト出力要求ができるものとする。なお第3.2.1項で詳述したように、コンパイラ出力のオブジェクトモジュールの保存は避けるべきであるが、やむを得ず保存する場合は第3.2.1項の注意事項に十分留意しなければならない。FORTC文による処理内容を図11に示す。

4.2.2 LIED文

各種状態のプログラムを結合編集し、1つのロードモジュールの作成を指示する文であり、ロードモ

ジュール状態の個人ファイルの自動呼出し入力と、メンバ名指定による入力を共に複数個可能とし、リンケージエディタ出力のロードモジュール保存の指示と、リスト出力要求とが可能なものとする。なお、やむを得ずオブジェクトモジュールを保存しているユーザに対して、その入力も複数個可能なものとするが、第3.2.1項の注意事項には十分留意しなければならない。LIED文による処理内容を図12に示す。

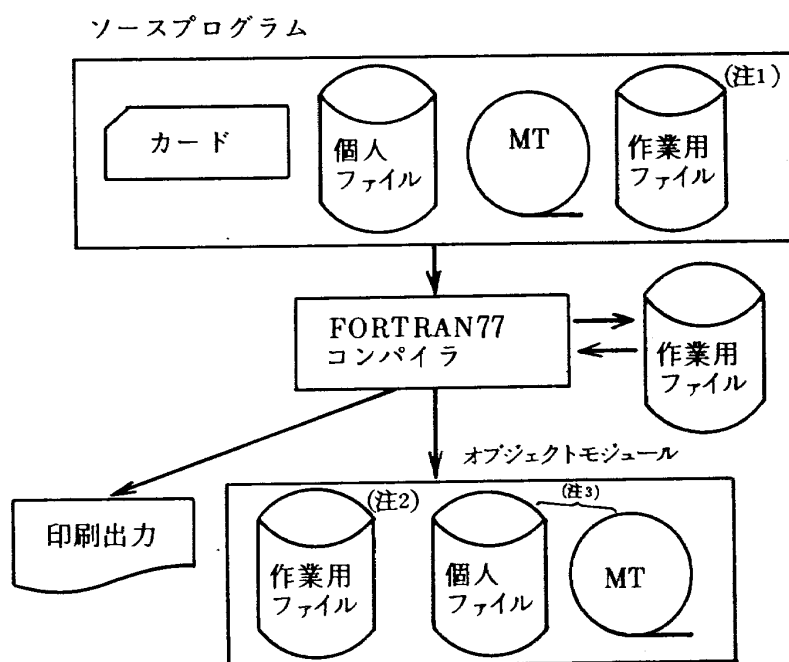
4.2.3 GO文

プログラムの実行を指示する文であり、実行プログラムの指定が可能であると共に、実行時データの入力が複数個可能なものとする。GO文による処理内容を図13に示す。

4.2.4 LINKGO文

プログラムの結合編集と実行とを1つのジョブステップで行うことを指示する文であり、^(注)コンパイラから引渡されるオブジェクトモジュールがない場合

(注) これを実現するために、リンケージエディタではなくローダを使用する^{7),8)}

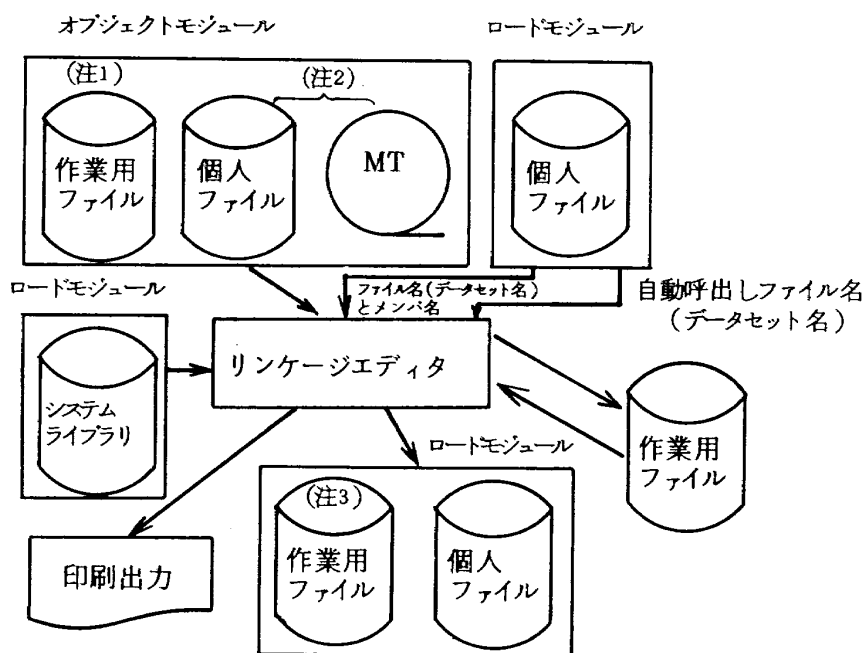


(注1) APTRANから受渡されるソースプログラム（第4.7.1項参照）。

(注2) リンケージエディタに引渡されるオブジェクトモジュール。

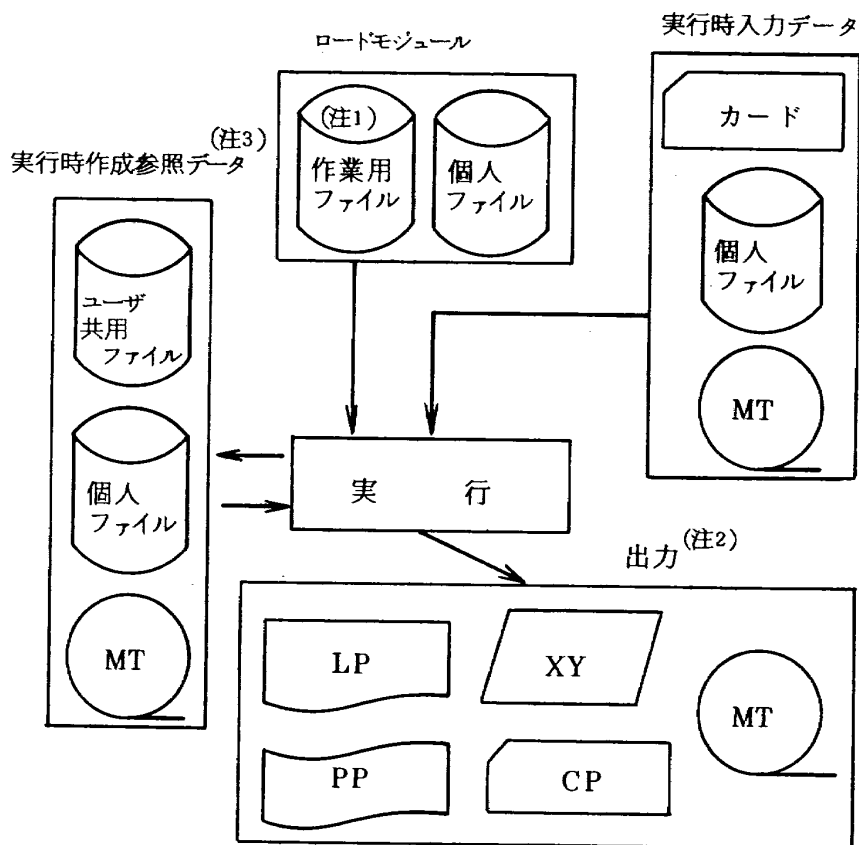
(注3) 第3.2.1項の注意事項参照のこと。

図11 FORTC文の処理フロー



- (注1) コンパイラから受渡されるオブジェクトモジュール。
 (注2) 第 3.2.1 項の注意事項参照のこと。
 (注3) 実行に引渡されるロードモジュール。

図 12 LIED文の処理フロー



- (注1) リンケージエディタから引渡されるロードモジュール。
 (注2) 第 4.3 項と第 4.7.3 項参照。
 (注3) 第 4.2.6, 7, 8 項参照。

図 13 GO文の処理フロー

にはロードモジュール状態の個人ファイルのメンバー名による入力を複数個可能とする。ただし、オブジェクトモジュールが引渡される場合にはこの入力できない。また、ロードモジュール状態の個人ファイルの自動呼出しと実行時データの入力とが共に複数個可能なものとする。なお、やむを得ずオブジェクトモジュールを保存しているユーザに対して、その入力も複数個可能なものとするが、第3.2.1項の注意事項には十分留意しなければならない。LINKGO文による処理内容を図14に示す。

4.2.5 ULIB文

第3.2.2項のロードモジュール作成法3に基づいたユーザライブラリの作成保存を指示する文であり、複数個のソースプログラム入力を可能とし、かつ出力

ロードモジュールの保存先ファイル(データセット)名は省略不可とする。ULIB文による処理内容を図15に示す。

4.2.6 USMT文

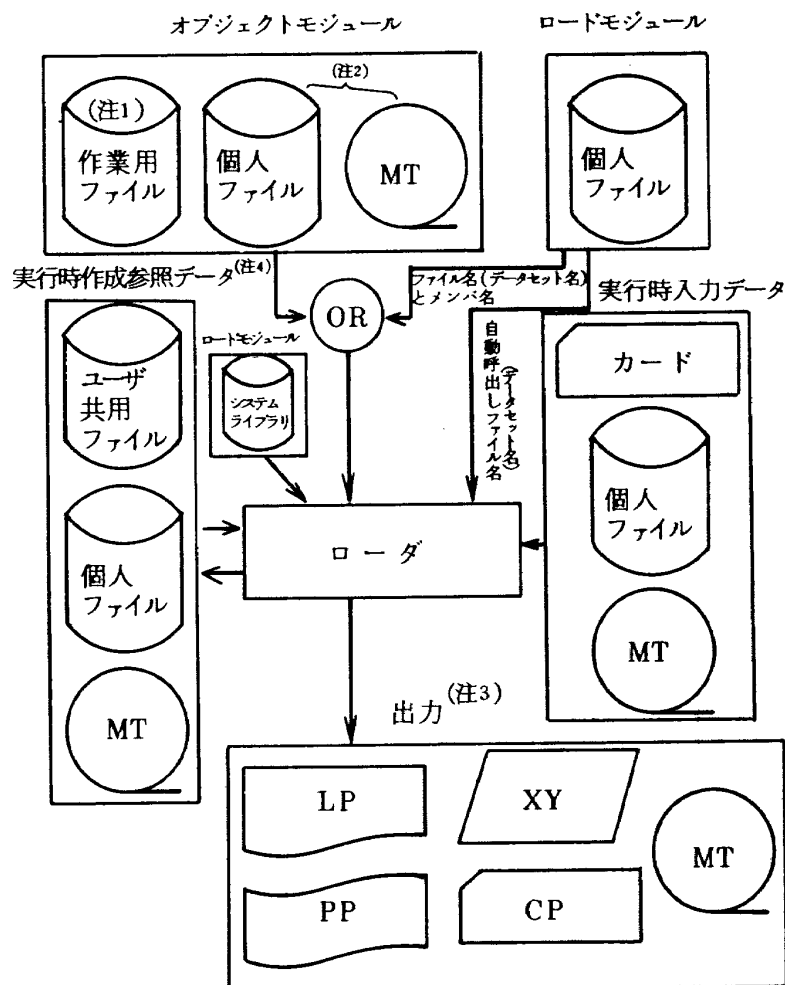
磁気テープファイル(データセット)の作成・参照を指示する文であり、SL形式、NL形式共に可能なものとする。

4.2.7 USDK文

個人ファイルの作成・参照を指示する文であり、作成されたファイル(データセット)は全て自動カタログされるものとする。

4.2.8 USWK文

ユーザ共用ファイルの使用を指示する文であり、ジョブ処理中に作成されたファイル(データセット)



- (注1) コンパイラから引渡されるオブジェクトモジュール。
 (注2) 第3.2.1項の注意事項参照のこと。
 (注3) 第4.3項と第4.7.3項参照。
 (注4) 第4.2.6, 7, 8項参照。

図14 LINKGO文の処理フロー

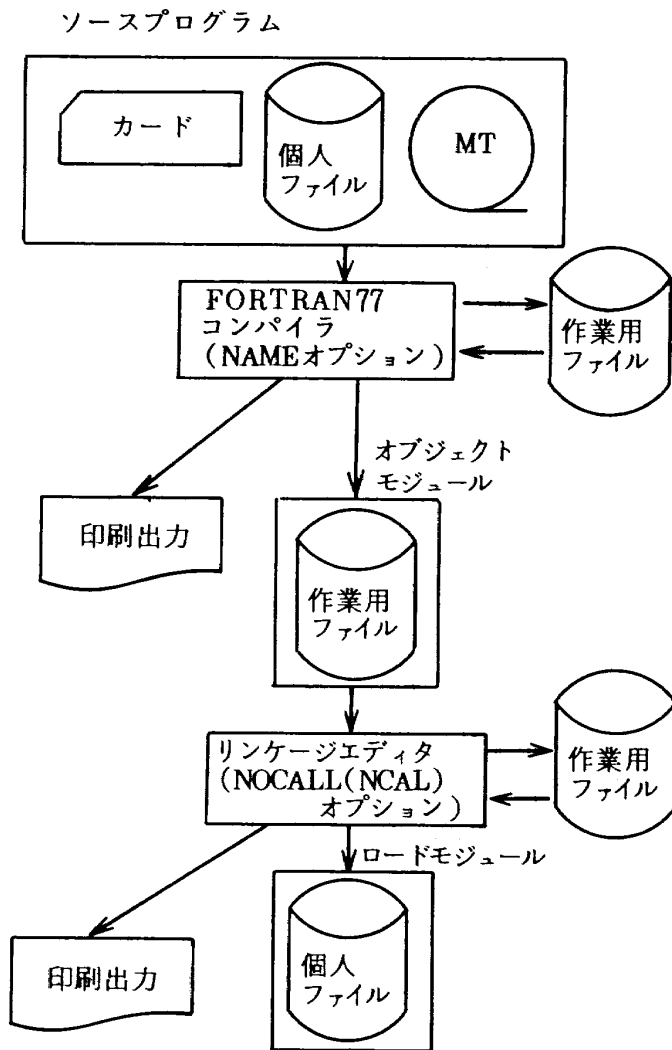


図 15 ULIB文の処理フロー

は全て、ジョブ終了と共に消去されるものとする。

4.2.9 CF文

入力ファイル(データセット)の連結を指示する文であり、いくつかの異なった装置から入力したファイル(データセット)を論理的に1つのファイル(データセット)として取扱うことを可能なものとする。

4.3 特殊出力要求用マクロ

標準出力(即ち、LP出力)の他に、特殊出力(即ち、CP, XY, PP出力)を指示するマクロが

必要である。

4.3.1 CP文

実行結果のカードパンチ出力を指示する文である。

4.3.2 PP文

実行結果のプリンタ・プロッタ出力を指示する文である。

4.3.3 XY文

実行結果のXYプロッタ出力を指示する文であり、使用する用紙とペンの種類の指定が可能なものとする。

4.3.4 PPXY文

実行結果をプリンタ・プロッタでモニタしてから、XY出力要求をするか否かを決定することを指示する文であり、XY出力時に使用する用紙とペンの種類が指定可能とする。

4.4 個人ファイル操作マクロ

個人ファイルの操作を大別すると、ソースプログラムやカードイメージのデータの内容更新と、ファイル(データセット)自体の作成・更新・複写・圧縮・消去とがある。これらの処理はTSS(会話型処理)で行うべきものであり、またTSS機能により容易に処理しえるものである。従って、個人ファイル操作マクロは本来不必要なものであるが、TSS装置が全て使用中である場合に、ファイル(データセット)内容の更新とファイル(データセット)の消去とがバッチ処理でも可能なように、下記のマクロを用意する。よって、このマクロの積極的な用途はない。

4.4.1 UPDATE文

カードイメージ(ソースプログラムまたはデータ)の個人ファイルの内容更新を指示する文であり、補助制御文を使用してレコードの挿入・置換・削除・識別順序番号付加が可能であり、またマージデータを使用してレコードの置換・追加・挿入が可能なものとする。

註) プロッタとしてはXYプロッタとプリンタプロッタ(PP)の2機種を考える。なお、ユーザプログラム上ではプロッタの機種を意識せずに、プロッタ共通のベーシック・ソフトウェア(PSP)を使用できるものとし、実行結果を出力する装置の選択はジョブストリーム上で対処できるようにする。即ち、ジョブストリームの中にPP文を挿入すればプリンタプロッタに出力でき、XY文を挿入すればXYプロッタに出力でき、PPXY文を挿入すればプリンタプロッタ出力後にXYプロッタ出力も可能とする。

4.4.2 DELETE 文

個人ファイルまたは個人ファイルのメンバの消去を指示する文である。

4.5 媒体変換用マクロ

各種の入力装置、記憶装置及び出力装置間でのファイル（データセット）の媒体変換を指示するマクロとして、以下のものが必要である。ただし、大型計算機システムでは、磁気テープ及びフロッピ・ディスクを使用する媒体変換はバッチ処理として取扱うよりも、ユーザコンソールのコマンド処理として取扱う方が断然優れている（詳細は第 6 章）。

4.5.1 CRTODK 文

カードから個人ファイルへの媒体変換を指示する文である。

4.5.2 CRTOMT 文

カードから磁気テープへの媒体変換を指示する文である。

4.5.3 CRTOLP 文

カードの内容のリスト出力を指示する文である。

4.5.4 CRTOCP 文

カードの複写を指示する文である。

4.5.5 DKTODK 文

個人ファイルから個人ファイルへの複写を指示する文である。

4.5.6 DKTOMT 文

個人ファイルから磁気テープへの媒体変換を指示する文である。

4.5.7 DKTOLP 文

個人ファイルの内容のリスト出力を指示する文である。

4.5.8 DKTOCP 文

個人ファイルの内容のカード出力を指示する文である。

4.5.9 DKTOFP 文

個人ファイルからフロッピ・ディスクへの媒体変換を指示する文である。

4.5.10 FPTODK 文

フロッピ・ディスクから個人ファイルへの媒体変換を指示する文である。

4.5.11 MTTODK 文

磁気テープから個人ファイルへの媒体変換を指示する文である。

4.5.12 MTTOMT 文

磁気テープから磁気テープへの媒体変換を指示する文である。

4.5.13 MTTOLP 文

磁気テープの内容のリスト出力を指示する文である。

4.5.14 MTCOPY 文

磁気テープから磁気テープへのデッドコピーを指示する文であり、第 4.5.12 項とは機能を異にする。

4.5.15 BACKUP 文

区分編成ファイル（データセット）のメンバを磁気テープへ退避することを指示する文であり、第 4.5.6 項とは大いに機能を異にする。^(註)

4.5.16 RESTORE 文

BACKUP 文で磁気テープ上に退避した区分編成ファイル（データセット）を個人ファイルへ復元することを指示する文であり、第 4.5.11 項とは大いに機能を異にする。

4.6 ファイル（データセット）情報出力用マクロ

個人ファイルに関する情報の出力を指示するマクロの他に、SL 形式磁気テープのラベル情報の出力を指示するマクロが必要である。

4.6.1 DKLIST 文

個人ファイルの使用状況の出力を指示する文であり、ファイル（データセット）名毎に、レコード形式、編成法、ブロック長、レコード長、確保容量、使用中容量、最新参照日などを出力するものとする。

4.6.2 DRTYLIST 文

区分編成ファイル（データセット）毎の全メンバ名の出力を指示する文である。

(註) MONITORⅦ システムでは磁気テープ上に順編成ファイルの他に、エレメント・セバレータ付き順編成ファイルが作成でき、かつエレメント名で磁気テープファイルの参照が可能である。OSⅣ システムでは磁気テープ上に順編成ファイルしか作成できない。また、BACKUP 文で退避した磁気テープ上の区分編成ファイルもメンバ名では参照できない。

4.6.3 MTLIST文

SL形式磁気テープのラベル情報の出力を指示する文であり、ボリューム毎にファイル（データセット）名、ファイル（データセット）順序番号、記録密度、レコード形式、ブロック長、レコード長などを出力するものとする。

4.7 特殊用途用マクロ

以上の他に、以下のような特殊用途用マクロを用意する必要がある。

4.7.1 APTRAN文

AP-FORTRANで書かれたソースプログラムをFORTRAN 77のソースプログラムに変換することを指示する文であり、入力ソースプログラムの入力指定ができ、変換制御データの入力指定が可能であり、かつ出力ソースプログラムの個人ファイルへの保存が可能なものとする。APTRAN文による処理内容を

図16に示す。

4.7.2 MTR文

磁気テープ上のデータの入力スプール処理を指示する文であり、この文で指定したファイル（データセット）はジョブ投入手続時にユーザ共用ファイル上に格納され、かつジョブ終了と共に消去されるものとする。

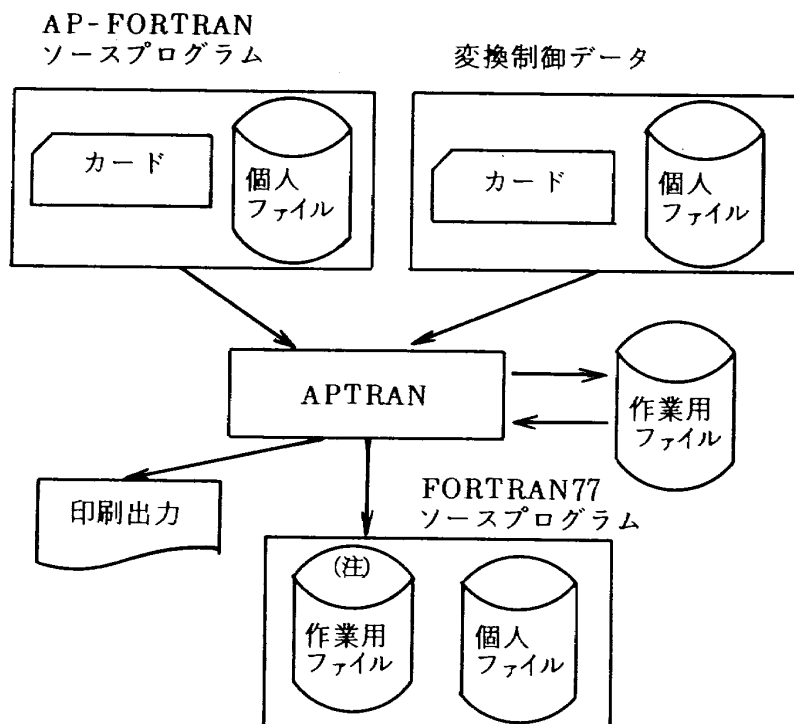
4.7.3 MTW文

磁気テープ上に作成するデータの出力スプール処理を指示する文であり、この文で指定したファイル（データセット）はジョブ実行中に一たんユーザ共用ファイルに格納され、ジョブ終了後ユーザによる磁気テープへの取り出しを待つものとする。^(註)

4.7.4 M7TOF4文

MONITOR VIIシステム用マクロで記述されたジョブの投入を指示する文であり、この文によりOS IVシステム用ジョブストリームを自動的に生成し、

(註) 現有システムではジョブの実行時に、磁気テープへ直接アクセスされる。このため、磁気テープを使用するユーザはジョブの実行に立合わなければならない、また磁気テープ装置の占有時間も徒に長い。次期システムではジョブ実行時における磁気テープへのアクセスを禁止し、上記の弊害を取りのぞく。MTR文とMTW文は磁気テープを使用するユーザに対して、ジョブの投入時と結果の取出し時にのみセンタにすればよいことを可能にするものである。



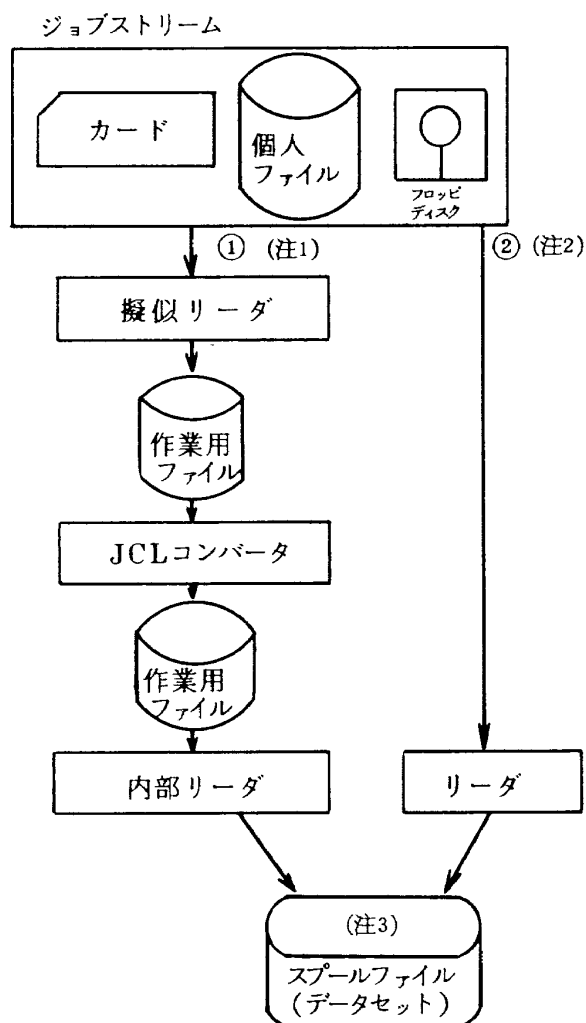
(注) FORTRAN77コンパイラに引き渡される
変換後ソースプログラム

図16 APTRAN文の処理フロー

正常なジョブの受付けを可能なものとする。ただし、投入可能マクロには制限があり、また OS IV システムの新規機能は利用できないことに注意する。できるだけ M7 TOF4 文を使用すべきでない。M7 TOF4 文による処理内容を図 17 の①に示す。

4.7.5 F4 TOX8 文

F4 システム用マクロで記述されたジョブを X8 システムへ投入することを指示する文であり、この文により X8 システムは自動的に X8 用ジョブストリームを生成し、正常なジョブの受付けを可能なものとする。ただし、投入可能なジョブはセンタで指定したマクロのみを使用し、かつ個人ファイルを使用していないものに限る。F4 TOX8 文による処理は図 17 に同様である。



(注1) ①はM7TOF4文のある場合。

(注2) ②はM7TOF4文のない場合。

(注3) ここにある処理待ちキューに登録されると、ジョブの受付け処理が終了する。

図 17 M7TOF4 文の処理フロー

4.7.6 HKJOB文, HJEND文

HKJOB文はX8システムからF4システムへのリモートバッチジョブであることを指示する文であり、ジョブストリームの先頭になければならず、HJEND文は最後になければならない。即ち、F4システム用ジョブストリームをHKJOB文とHJEND文とではさめば、X8システムがジョブをF4システムへ引渡すことを可能なものとする。

4.8 マクロ間の相互関連について

マクロ間の相互関連は次の仕様を満たすものとする。

(i) 保存をしなければ、APTRAN文の出力（FORTRAN 77ソースプログラム）はその入力を指示した最初の FORTC 文へ自動的に引渡され、FORTC 文の処理終了とともに消去されるものとする。但し、保存をした場合には、その APTRAN 文と後続マクロとの関連性は一切なくなるものとする。

(ii) 保存をしなければ、FORTC 文の出力（オブジェクトモジュール）は最初に現われる LIED 文または LINKGO 文へ自動的に引渡され、LIED 文または LINKGO 文の処理終了とともに消去されるものとする。保存をした場合には、その FORTC 文と後続マクロとの関連性は一切なくなるものとする。なお FORTC 文で特に指示がなければ、APTRAN 文の出力の自動受渡しはないものとする。

(iii) LIED 文の出力（ロードモジュール）はそれを保存する場合も保存しない場合にも、後続の GO 文へ自動的に引渡されるものとする。なお、LIED 文で特に指示すれば、先行マクロとの関連性を一切なくし、その LIED 文から新たな処理の開始が可能なものとする。

(iv) LINKGO 文で特に指示すれば、先行マクロとの関連性を一切なくし、その LINKGO 文から新たな処理の開始が可能なものとする。その LINKGO 文は後続マクロとの関連性が一切ないものとする。

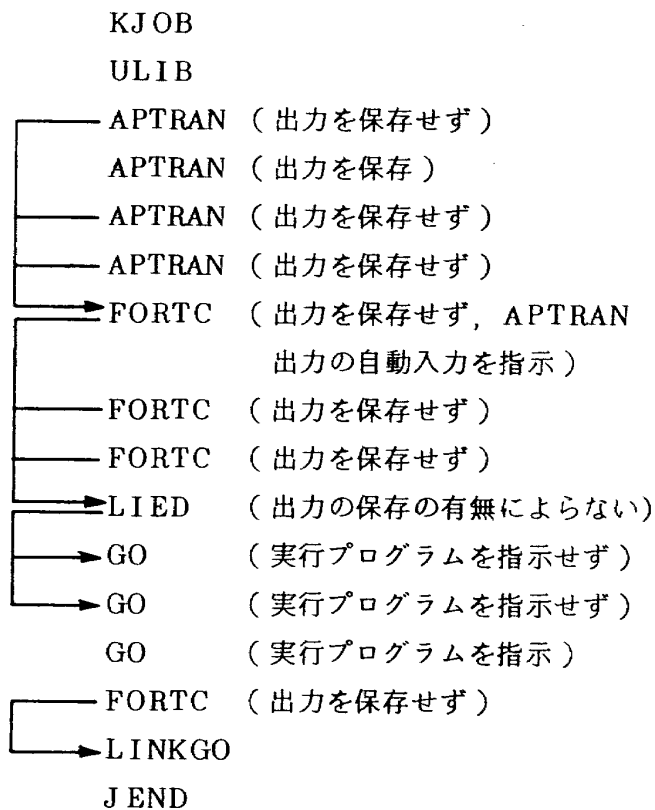
(v) GO 文で実行プログラムの指定がなければ、直前の LIED 文の出力（ロードモジュール）が実行され、実行プログラムの指定があれば、そのプログラムが実行されるものとする。

(vi) NOCALL(NCAL)オプションの指定がなけ

れば、マクロ内で起動されるリンケージエディタまたはローダは常にシステムライブラリを自動呼出し、ユーザがシステムライブラリの呼出しを意識する必要のないものとする。

(vii) ULIB文および各種媒体変換文は前後のマクロとの関連性が一切なく、それぞれ処理が独立しているものとする。

(viii) 以上の仕様に基つくと、下記の例に示すように、各マクロ間の関連性が簡明化されるから、ジョブストリームが容易に記述できる。ここで、矢印はマクロ間で自動的に入出力の受渡しがなされる関係を示している。



なお、第二APTRAN文の出力をもFORTC文の入力としたい場合には、第一、第二、第三のいずれかのFORTC文の入力パラメータでその保存先ファイル名(データセット名)を明記すればよい。

5. X8システム用マクロの設計

OSIV/X8システムは小型から中型までの計算機システムに適したOSといえる。特に小型計算機システムの場合、ユーザがシステムを占有し、直接計算機を操作することが頻繁となる。このため、バッチ処理依頼の手続きや、媒体変換機能にはできるだけ柔軟性を持たせておく必要がある。これをマクロ

の観点から言い直すと、多少マクロの記述形式が複雑になっても、各マクロには最大限の機能を持たせておく必要があり、また多種類のマクロを用意しておいた方がよい。

以上のことを念頭において、本章では前章の仕様を満たしうるX8システム用マクロを実際に設計する。

なお、小型計算機システムとしてはカード読取り装置、ラインプリンタ装置、XYプロッタ装置、磁気ディスク装置、磁気テープ装置、TSS装置を装備し、かつ大型計算機システムと回線結合されているという構成を想定している。

5.1 ジョブ制御マクロ機能

X8システムのジョブ制御マクロ機能はMONITORⅦシステムでの機能を包含しており、かつ有用な新規機能を有している。従って、かなり融通性に富んだマクロの作成が可能である。

5.2 全体的注意事項

5.2.1 ファイル名について

(1) 個人ファイルのファイル名

個人ファイルのファイル名は26文字以内であり、かつ次の条件を満たしていなければならない。

- ・第1修飾子は規約通りの名前であること。
- ・ピリオドは連続してはならない。
- ・ピリオドに続く文字は英字で始まる8文字以内の英数字であること。
- ・ピリオドで終ってはいならない。

(2) MTファイル名

MTファイル名は17文字以内で、かつ第1修飾子の規約を除く(1)の条件と下記の条件を満たしていなければならない。17文字をこえるファイル名を指定した場合には先頭から17文字のみが有効となる。

- ・ファイル名の先頭は英字で始まる11文字以内の英数字であること。

5.2.2 記述上の注意

(1) 一般的事項

各マクロのパラメータ記述形式は第5.3項で示す書き方に従わなければならない。なお、記述上の全体的取り決めは次の通りである。

- []で示したパラメータは省略可能である。{ }で示したパラメータはその中の一項目のみを選択する。(項目, …)は項目をくり返し指定可能なことを意味する。
 - 位置パラメータは指定された順番に記述しなければならない。キーワードパラメータの記述順序は位置パラメータの後であれば順不同でよい。^(注1)なお、キーワードパラメータの右辺に現われるサブパラメータは示されている順序で、必要なものを記述すれば、間違うことがない。
 - 全てのマクロ文は次のカードに継続できるが、その際、必ず“, ”記号で記述が終っていなければならない。また継続行の第一桁は空白でなければならない。なお、注釈以外の継続では継続欄の記入を省略できる。
- (2) 特殊事項
- 第 5.3 項で示す記法の内、一部のみ次の書き方も可能である。
- 一番外側の括弧記号の代わりに、引用符記号を使用してもよい。また、一番外側の括弧内に記述できる文字数は 128 個である。
 - 特殊記号“, ”, “:”および“+”を含む文字列の括弧は省略できる。例えば、(H01.ABC)は単にH01.ABCと記述することができる。
 - 個人ファイルのファイル名は一般に(ファイル名, [パスワード], ファイル名, [パスワード], ……)のように、ファイル機密保護のためのパスワードを含めて指定しなければならない。従って、単一のファイル名の指定は(ファイル名, [パスワード])となるが、もしこのファイルをパスワード保護していないならば括弧を省略してもよい。即ち、(H01.ABC,)でも、単にH01.ABCでもよい。また、多ファイルの記述の場合でも、最後のファイル名に対するパスワードを省略したときには最後の“, ”は省略できる。
 - キーワードパラメータの右辺のサブパラメータの

指定が一個であれば、括弧を省略してもよい。

5.2.3 個人ファイルの作成・参照について

(1) 保存時の注意

ソースモジュール^(注2)とオブジェクトモジュールは固定長形式で管理され、ロードモジュールは不定長形式で管理される。このため、一つのファイルの中に両者を混在させてはならない。

(2) ファイルの自動カタログ・アンカタログ

システムが自動的に、ファイル作成時にカタログ登録し、ファイル消去時にアンカタログする。従って、ユーザは各自のファイルの作成・参照・消去時に、装置のボリューム通番とグループ名を何ら意識する必要がない。

(3) 定義可能なアクセス名U00～U99の内、U05はカードの読み込み、U06はリストの出力、U98～U99はXY出力のためにシステムが使用する。なお、U01もシステムが使用する場合がある。

(4) ファイルの作成時に、レコード長、ブロック長、レコード形式を省略すると、システムの標準値に基づいてファイルが作成される。従って、作成時に上記パラメータを省略したならば、参照時にもこれらパラメータを省略すること。

(5) ファイルを機密保護する際に指定するパスワードは4文字の英数字でなければならない。

5.2.4 マクロの設計について

(1) 不要マクロ

第4章で記述した各種マクロの内、次のものはX8システムでは不要である。

CP文, PP文, PPXY文, CRTOCP文, DK-TOFP文, FPTODK文, APTRAN文, MTR文, MTW文, M7TOF4文

(2) 追加マクロ

第4章で記述したマクロ以外に、X8システムの利用の便宜のため、次のマクロを追加する。なお、それぞれの設計仕様は各項で説明する。

SAME文, INSERT文, MERGE文, OMIT文,

(注1) 下記の例で示すと、①が位置パラメータ、②がキーワードパラメータである。

¥KJOB ユーザ登録名, TIME=実行打切時間

①

②

(注2) ソースモジュールとはカードイメージのファイルを意味し、その内容はソースプログラムまたは書式付データである。

RENUM文

(3) マクロ間の相互関連

X8 システムでは FORTC 文と GO 文を、第 4.8 項の(ii)と(V)の設計仕様を満たしたものを実現できないため、次のような対策を講じる。

- FORTC 文が複数あっても、それらが作成するオブジェクトモジュールの内、最後に作成したものしか LIED 文の入力とならない。このための対策として、FORTCF 文を用意する（詳細は FORTCF 文を参照）。
- GO 文が複数あっても、LIED 文が作成するロードモジュールは最初の GO 文にしか引渡されない。このための対策については GO 文の項で詳述する。一方、X8 システムには FORTC 文および LIED 文の出力の保存の有無にかかわらず、それぞれ直後の LIED 文または GO 文へ自動的に出力の受渡しができる機能をそなえている。X8 用マクロではこの機能を採用して、ジョブストリーム中の FORTC 文と LIED 文、LIED 文と GO 文間の関連性を強化する。

5.3 各マクロの設計

5.3.1 KJOB 文

(1) 記述形式

表 5.1

(2) 機能上の特記事項

(2.1) ユーザ登録名^(註)、ジョブクラス、HOLD、OHOLD は位置パラメータであるから、この順序で記述しなければならない。

(2.2) USER クラスを指定した場合には、自動的に HOLD パラメータがセットされる。

(3) 使用例

(3.1) 位置パラメータの内、HOLD パラメータのみを省略する場合には次のように記述する。

```
¥ KJOB H01XPASS, DATA, , OHOLD
```

(3.2) 位置パラメータの内、OHOLD パラメータのみを省略する場合には次のように記述する。

```
¥ KJOB H01X PASS, SHRT, HOLD
```

(3.3) 位置パラメータの内、HOLD、OHOLD パラメータを省略する場合には次のように記述する。

```
¥ KJOB H01 X PASS, USER, TIME= 10
```

5.3.2 JEND 文

(1) 記述形式

表 5.2

(2) 機能上の特記事項

本マクロはパラメータを有しない。必要ならば、パラメータ記述欄に注釈を記入できる。

5.3.3 FORTC 文

(1) 記述形式

表 5.3

(2) 機能上の特記事項

(2.1) X8 システムには第 4.8 項の(ii)の仕様を満たす機能がない。このため、2 枚目以後の FORTC 文によるオブジェクトモジュールも LIED 文の入力としたい場合には、2 枚目以後は FORTCF 文（第 5.3.4 項参照）を使用しなければならない。

(2.2) SF パラメータに区分編成ファイルを指定する場合には、メンバ名はコンパイラオプションの ELM オプションで指定する。このとき、PARAM = (ELM(メンバ名, ...)) と指定すると、指定したメンバだけが参照され、PARAM = (ELM(*)) と指定すると、指定ファイルの全メンバが参照される。なお、ELM オプションで指定できるメンバの個数は最大 10 個である。

(2.3) SF パラメータは多ファイルの指定が可能であるが、その際、全ファイルの編成、レコード形式、ブロック長、レコード長は同じでなければならない。また、ファイルの個数に制限はないが、一番外側の括弧内に記述できる文字、記号の数は 128 文字以内でなければならない。

(註) ユーザ登録名は 8 文字で表現し、形式は次の通りとする。

ユーザ登録名 ; H01XPASS

①②③ ④

ここで、①は部識別を、②は部内ユーザ番号を、③は投入元の識別（英数字又はピリオド）を、④はパスワードを表わす。なお、先頭の 3 文字をユーザ名と呼ぶ。

(2.4) SFパラメータで区分編成ファイルを複数個指定したとき、同名のメンバが複数のファイルに存在すると、SFパラメータで先に指定したファイルのメンバが有効となる。

(2.5) ソースプログラムの入力媒体は、カード、DKまたはMTを可能とする。ただし、一枚のFORTC文には一種類しか選択できない。従って、複数媒体を入力する場合には、FORTC文とFORTCF文を複数定義する。

(2.6) カタログ登録してあるMTボリューム上のファイルを参照する場合にも、SVパラメータは省略できない。省略した場合にはDKボリューム上のファイルと見なされる。

(2.7) SFパラメータは多ファイル、SVパラメータは多ボリュームの指定が可能であるが、その組合せは単ファイル単ボリューム、単ファイル多ボリューム、多ファイル単ボリュームでなければならない。多ファイルで多ボリュームの指定はできない。ただし、不特定ボリューム(SLMT)を指定する場合には、多ファイル多ボリュームの指定が可能である。なお、不特定ボリュームの指定は常にSV = SLMTとしなければならない。また、SV = (ボリューム通番, ボリューム通番, …)の記述を省略して、先頭のボリュームのみ指定するだけでもよい(詳細はUSMT文参照)。

(2.8) 印刷出力を個人ファイルに格納する場合には、LFパラメータでファイル名を指定する。新規作成ファイルのときにはLTRKパラメータを省略できない。また、ファイル編成は順編成に限る。また、パスワード保護の指定ができない。

(2.9) FORTRAN 77 コンパイラオプションはPARAMパラメータで指定できるが、次の事項に注意する。

(i) NAMEオプションについては第3.2項を参照のこと。

(ii) GOオプションを指定すると、一つのジョブステップで翻訳から実行までを処理することができる。ただし、オブジェクトモジュールもロードモジュールも保存することはできず、また、実行ロード

モジュールの分だけユーザリジョンが増大する。

(iii) LINECOUNT(0)を指定すると印刷情報はすべてベタ打ち(1頁に66行印刷)となり、プログラムのヘッダリストは最初に1回だけ出力され、そのあとは編集のための行送りと改頁は行わないのでリスト用紙の節約ができる。

(iv) コンパイラオプションには、プログラム内の*PROCESS行^(注1)でも指定可能なものもあるが、両方で相矛盾するオプションを指定した場合は、*PROCESS行での指定が有効となる。

(v) GOオプションを指定した場合、"/"で区切れば、コンパイラオプションの後にロード及び実行時オプションを指定できる(使用例参照)。

(vi) SINCパラメータで指定するファイルは区分編成ファイルでなければならない。また、*INCLUDE行^(注2)で指定するメンバをソースプログラムの中に組込む場合には、そのファイル名はSINCパラメータで指定しなければならない。なお、SINCパラメータを指定すると、同時にINCLUDEオプションを指定したことになるので、PARAMパラメータで改めて指定してはならない。また、SINCパラメータを省略すると、*INCLUDE行が出現しても注釈行と見なされる。

(vii) コンパイラチェックのみを行う場合、NOBJECTオプションを指定すると、コンパイラはオブジェクトモジュールをディスク上に作成しないから、IO頻度が少く、ジョブの経過時間を短縮できる。

(viii) ソースプログラムのリスト出力が不要な場合には、NOSOURCEオプションを指定する。

(2.10) コンパイラの出力(オブジェクトモジュール)は出来るだけ保存しないこと(第3.2.1項参照)。やむを得ず保存する場合にのみ、以下の事項が有効である。

(i) オブジェクトモジュールの出力媒体はDKとMTを可能とする。ただし、個人ファイルに保存した場合のみ自動的にLIEDの入力となる。

(ii) DISPパラメータは、RVパラメータに対して有効とする。

(注1) ソースプログラム中に記述する文であり、一部のコンパイラオプションを指定するオプション行である。

(注2) ソースプログラム中に記述する文であり、区分編成ファイルのメンバの内容をプログラム中に組み込むオプション行である。

(iii) PSWとTRKパラメータは、RFパラメータで指定するファイルに対して有効とする。

(iv) RFパラメータのみ指定した場合には順編成ファイルとなる。RMEMパラメータも指定すれば区分編成ファイルとなる。

(v) オブジェクトモジュールを新規作成ファイルに保存する場合、直後のジョブステップで参照または追加作成のためこのファイルを指定すると、カタログ登録がまだ完了していないためにエラーとなる。

(vi) ファイルの新規作成時にはTRKパラメータの指定を省略できない。

(3) 使用例

(3.1) ソースプログラムをカード入力し、コンパイル結果を印刷出力する。

```
¥ FORTC
```

```
  ソースプログラムカード
```

(3.2) 順編成の個人ファイルを参照してコンパイルする。

```
¥ FORTC SF=(H01.A,P001)
```

(3.3) 区分編成の個人ファイルの全メンバを参照してコンパイルする。

```
¥ FORTC PARAM=(ELM(*)),  
SF=H01.E
```

(3.4) 順編成の個人ファイルを3つ入力してコンパイルする。

```
¥ FORTC SF=(H01.A,P001,  
H01.B,,H01.C,P001)
```

(3.5) 区分編成の個人ファイル3つに含まれる3つのメンバを入力してコンパイルする。

```
¥ FORTC PARAM=(ELM(M1,M2,  
M3)),SF=(H01.E,,H01.F,,  
H01.G)
```

(3.6) MTファイル(単ファイル単ボリューム)を入力してコンパイルする。

```
¥ FORTC SF=MTFILE,  
SV=NM0001
```

(3.7) MTファイル(単ファイル多ボリューム)を入力してコンパイルする。

```
¥ FORTC SF=H01.MT,  
SV=(NM0001,NL0030)
```

(3.8) MTファイル(多ファイル単ボリューム)を入力してコンパイルする。

```
¥ FORTC SF=(MF1,MF2,MF3),  
SV=NS0050
```

(3.9) GO, NOSOURCE オプションを指定し、実行時使用データをソースプログラムカードと一緒に入力する。この場合、プログラムとデータを入力区切り文(¥/)で区切る。なお、入力データ区切り文はリスト上には出力されない。

```
¥ FORTC PARAM=(GO,  
NOSOURCE)
```

```
  ソースプログラムカード
```

```
¥/
```

```
  入力データカード
```

(3.10) コンパイラオプション(GOとMAP)と実行時オプション(DATAON)を指定する。

```
¥ FORTC PARAM=(GO,  
MAP//DATAON=8)
```

```
  ソースプログラムカード
```

(3.11) *INCLUDE 行で組込むメンバが含まれるファイルを指定する。このとき、PARAMパラメータの中でINCLUDEオプションを指定してはならない。

```
¥ FORTC SINC=(H01.E,,H01.F,  
P003,H01.G)
```

```
  ソースプログラムカード
```

(3.12) オブジェクトモジュールを区分編成で保存し、それをWRITE保護ファイルとする

```

¥ FORTC RF=H01.A, RMEM=M1,
    PSW=(PS00,WRITE),
    TRK=(10,2)

```

ソースプログラムカード

(3.13) 下の例において、①と②は単なるコンパイルチェックのみとなり、③のみが LIED の入力となる。①、②及び③のオブジェクトモジュールを全て LIED の入力としたい場合は、②と③は FORTCF 文（次項参照）を使用しなければならない。

```

¥ FORTC                                     ①
    ソースプログラムカード
¥ FORTC SF=H01.A                           ②
¥ FORTC SF=MF5,SV=SLMT                     ③
¥ LIED

```

5.3.4 FORTCF 文

X8 システムには、第 4.8 項の(iii)の仕様を満たす機能がないために、この文を特に用意し、複数のコンパイルステップの出力を全て連結して、リンケージエディタの入力としえるようにする。

(1) 記述形式

表 5.4

(2) 機能上の特記事項

(2.1) 先行ジョブステップに FORTC 文がなければならぬ。

(2.2) FORTCF 文の展開形には、オブジェクトモジュールの出力先ファイル定義文はなく、先行の FORTC 文におけるファイルを引き継ぐ。従って、FORTCF 文が出力したオブジェクトモジュールは FORTC 文が出力したオブジェクトモジュールに連結される。また、FORTCF 文の後に FORTC 文が現れると、新しいファイルが再定義され、先に連結されたオブジェクトモジュールは以後一時ファイルとして参照できない。

(3) 使用例

下の例において、①、②および③のオブジェクトモジュールは全て LIED の入力となる。

```

¥ FORTC                                     ①
    ソースプログラムカード
¥ FORTCF SF=(H01.SUB,                      ②
    P001)
¥ FORTCF SF=MTFILE,                        ③
    SV=SLMT
¥ LIED

```

5.3.5 LIED 文

(1) 記述形式

表 5.5

(2) 機能上の特記事項

(2.1) このマクロは静的結合のときのみ使用できる。^(注)動的結合（オーバレイ、動的プログラム、ダイナミックリンク）を使用する場合にはユーザ各自が展開形のジョブ制御文を作成しなければならない。

(2.2) ALIB パラメータで指定できるファイルの作成法については第 3.2.2 項を参照のこと。ALIB 指定のファイルは自動呼出しライブラリとして参照され、その中に外部参照と一致するメンバ名があれば自動的に結合編集される。従って、ALIB 指定ファイルでは、メンバ名をパラメータで指定する必要はない。ただし、初期値設定副プログラム（BLOCKDATA）を自動呼出しライブラリの対象とするためには、引用するプログラムの中で、EXTERNAL 文でそのメンバ名を指定しなければならない。なお、システムライブラリは自動的に参照されるので、ユーザは各自のライブラリのみを意識すればよい。

(2.3) IEF と IEMEM パラメータは S = OFF の場合、LIED への追加入力を意味する。一方、S = ON の場合には LIED への主要入力を意味する。主要入力のメンバ名と追加入力のメンバ名に同一のものがあると、主要入力の方が有効となり、追加入力の方は無効となる。全てのメンバ名が異なっていれば、全てのメンバが有効となる。主要入力（または、追加入力）の中で同一メンバ名があると、先に

(注) 静的結合とは、処理する一連のプログラムを一個のロードモジュールにしてから実行に入る方式のことである。

入力された方が有効となり、後から入力されたものは無効となる。なお、主要入力と追加入力だけでは外部参照関係が解決されていない場合、システムは自動的に、ユーザ指定のALIBを調べる。

(2.4) リンケージエディタが作成するロードモジュールは区分編成で管理される。レコード形式は不定長であり、ブロック長はLMSIZEオプションで指定できるが、指定しない場合には3KB長となる。また、作成は追加(AD)モードで行われるので、ADサブパラメータは指定する必要はない。ただし、追加出力する際、ブロック長が既存ファイルのそれと同じかまたはそれ以下でなければならない。

(2.5) 出力ロードモジュールを保存するとき、OEMEMパラメータを指定しなければならない場合と、指定してはならない場合とがある。コンパイラジョブステップでNAMEオプションの指定をせずにオブジェクトモジュールを作成している場合にはメンバ名を指定しなければならない。NAMEオプションの指定をしている場合にはメンバ名を指定してはならない。また、新規作成の場合、TRKパラメータの指定を省略することはできない。

(2.6) オブジェクトモジュールは出来るだけ保存すべきでない(第3.2.1項参照)。やむを得ず保存している場合にのみ、以下の事項が有効である。

(i) RFパラメータは多ファイルの指定が可能であるが、その際、全ファイルの編成、レコード形式、ブロック長、レコード長は同じでなければならない。また、区分編成ファイルの場合には、RMEMパラメータでメンバ名を指定しなければならない。

(ii) MTファイルの場合、RFとRVパラメータの組合せの内、多ファイルで多ボリュームの指定は不可能である(FORTC文参照)。

(3) 使用例

(3.1) 先行コンパイルジョブステップの出力したオブジェクトモジュールを主要入力して、結合編集を行いロードモジュールを一時ファイルとして出力する。

```
¥ LIED
```

(3.2) 先行コンパイラジョブステップの出力したオブジェクトモジュールを主要入力とし、既存の

ロードモジュールを追加入力して結合編集を行う。これにより、新しいロードモジュールに作り直すことができる。

```
¥ LIED IEF=H01.SUB2,  
      IEMEM=M1
```

(3.3) MTボリューム上の既存オブジェクトモジュールを主要入力し、既存のロードモジュールを追加入力して結合編集を行い、新しいロードモジュールを既存メンバと置き替える。

```
¥ LIED S=ON, RF=(FILE1,  
      FILE2, FILE3), RV=NL0005,  
      IEF=(H01.S, , H01.T),  
      IEMEM=(M1, M2, M3),  
      OFF=H01.U, PSW=P001,  
      OEMEM=(MEM20, R)
```

(3.4) ロードモジュールを主要入力とし、かつ、ロードモジュールの個人ファイルを自動呼出して、実行プログラムを作成する。

```
¥ LIED S=ON, IEF=(H01.F,  
      P001), IEMEM=M1,  
      ALIB=H01.F
```

5.3.6 GO文

(1) 記述形式

表 5.6

(2) 機能上の特記事項

(2.1) X8システムには第4.8項の(V)の仕様を満たす機能がない。即ち、LIED文が作成したロードモジュールは最初のGO文にしか受け渡しができない。2枚目以後のGO文に受け渡す必要のある時はロードモジュールを個人ファイルに保存し、2枚目以後のGO文でその実行プログラムの指定をしなければならない。

(2.2) GO文で定義する実行時使用データはプログラム中の入力文でファイル識別番号を5としなければならない。即ち、カード、MT、個人ファイル全ての場合について、READ(5, ...)とする。

(2.3) CONDパラメータを省略した場合、システムはCOND=(10, 49)を設定する。

(2.4) 実行時使用データのファイルは DF, DF1, DF2, DF3, DF4, の順に 5 個まで指定できる。区分編成ファイルの場合、それぞれ、DMEM, DMEM-1, DMEM2, DMEM3, DMEM4 が対応し、各々多メンバの指定ができる。ただし、多ファイルを指定する際、全ファイルの編成、レコード形式、ブロック長、レコード長は同じでなければならない。

(2.5) FORTC 文の(2)の(2.8)項参照。

(3) 使用例

(3.1) FORTRAN コンパイラジョブステップ、結合編集ジョブステップにつづいて GO ジョブステップを処理する。

¥ FORTC

ソースプログラムカード

¥ LIED

¥ GO

入力データカード

(3.2) ロードモジュールを既存の個人ファイルから入力する。

¥ GO EF=(H01.L, PS01),
EMEM=M1

(3.3) 実行時使用データを個人ファイル(順編成)から入力する。

¥ GO DF=(H01.DATA, PS01)

(3.4) 実行時データを MT ボリューム上から入力する。

¥ GO DF=DATA10, DV=NM0003

(3.5) 印刷出力の頁打切値を指定する。

¥ GO PAGE=200

(3.6) 実行時使用データを個人ファイル(区分編成)の複数メンバを連結して参照する。さらに COND パラメータで先行ジョブステップの完了コードが 20 以下であれば GO ジョブステップを実行することを指示する。

¥ GO DF=(H01.DATA, PS01),
DMEM=(M1, M2, M3), COND=20

(3.7) 実行時使用データとして、個人ファイル(順編成)を複数個指定する。

¥ GO DF=H01.D1, DF1=H01.D2,
DF2=H01.D3

(3.8) 同じプログラムをデータだけかえて複数ステップ実行する。

¥ FORTC SF=(H01.A,, H01.B,,
H01.C)

¥ LIED OEF=H01.F, OEMEM=M10

¥ GO

入力データカード

¥ GO EF=H01.F, EMEM=M10

入力データカード

¥ GO EF=H01.F, EMEM=M10

入力データカード

5.3.7 LINKGO 文

(1) 記述形式

表 5.7

(2) 機能上の特記事項

(2.1) FORTC 文が先行していない場合、EF パラメータでロードモジュールの個人ファイルをローダの主要入力とする指定ができる。ただし、FORTC 文の出力をこの文の入力とする場合にはこのパラメータを指定してはならない。入力ファイルは EF, EF1, EF2, EF3, EF4 の順に 5 個まで指定でき、それぞれ EMEM, EMEM1, EMEM2, EMEM3, EMEM4 が対応し、各々多メンバの指定ができる。

(2.2) " / " で区切れば PARAM パラメータで、ローダオプションと実行プログラムオプションを同時に指定することができる。ただし、実行プログラムオプションのみ指定の場合には " / " から始めなければならない。

(2.3) 本マクロを使用した場合、ロードモジュールを保存する指定はできない。

(2.4) 本マクロは FORTC の GO オプションでは仮想記憶が大きすぎる場合や、個人ファイルを自

動呼出ししたい場合に有効である。

(2.5) コンパイラの出力(オブジェクトモジュール)は出来るだけ保存しないこと(第3.2.1項参照)。やむを得ず保存する場合にのみ、オブジェクトモジュールの個人ファイルをローダの主要入力とする指定ができる。ただし、FORTC文の出力をこの文の入力とする場合にはRFパラメータを指定してはならないし、RFパラメータを指定したときはEFパラメータを指定することができない。

(2.6) FORTC文の(2)の(2.8)項参照。

(3) 使用例

(3.1) 先行コンパイラジョブステップが出力するオブジェクトモジュールを主要入力とし、ローダで結合編集と実行を行い、同時にロードしたプログラムのマップを出力する。

```
¥ LINKGO PARAM=MAP
```

入力データカード

(3.2) 既存のロードモジュールを主要入力とし、自動呼出しライブラリを参照し、ローダで結合編集と実行を行う。

```
¥ LINKGO EF=(H01.E,PS01),
      EMEM=M1,ALIB=(H01.SUB,
      PS02)
```

(3.3) 既存のソースプログラムを翻訳し、そのオブジェクトモジュールを主要入力とし、自動呼出しライブラリを参照し、さらに、実行時データファイルを指定して、ローダで結合編集と実行を行う。

```
¥ FORTC SF=(H01.A,,H01.B)
¥ LINKGO ALIB=(H01.L1,,
      H01.L2),DF=H01.DATA
```

5.3.8 ULIB文

(1) 記述形式

表 5.8

(2) 機能上の特記事項

(2.1) コンパイラのNAMEオプションはマクロの中で自動的に指定されるので、FPARAMパラメータでNAME、NONAMEオプションを指定してはならない。

(2.2) リンケージエディタのNOCALLオプションはマクロの中で自動的に指定されるので、LPA-RAMパラメータで、NOCALLオプションを指定してはならない。

(2.3) ロードモジュールの保存先ファイル名は必ず指定しなければならない。保存に当り、既に同じメンバ名がある場合は置き換えとなり、ない場合は追加となる。なお、出力は追加(AD)モードであるので、ADサブパラメータは指定する必要はない。

(2.4) 作成されたロードモジュールには自動的に下記のメンバ名が付加される。従って、メンバ名の指定をしてはならない。

プログラム単位	メンバ名
主プログラム PROGRAM文なし PROGRAM pgm	MAIN pgm
手続き副プログラム SUBROUTINE FUNCTION proc proc	} proc
初期値設定副プログラム BLOCKDATA BLOCKDATA blk	BLOCKD blk

(2.5) 本マクロを使用して作成したロードモジュールは、メンバ名指定入力としても、自動呼出しライブラリとしても使用できる。

(2.6) 本マクロはその前後にあるマクロとは完全に独立している(第4.2.5参照)。

(3) 使用例

既存の3つのソースプログラムファイルの全メンバを入力し、ユーザライブラリを新規作成する。

```
¥ ULIB FPARAM=(ELM(*)),
      SF=(H01.E,P001,H01.F,,
      H01.G),EF=H01.E10,
      TRK=(10,2)
```

5.3.9 USMT文

(1) 記述形式

表 5.9

(2) 機能上の特記事項

(2.1) アクセス名の形式はUnn又はFTnnF001

である。nn は 00 ～ 04, 07 ～ 99 の 2 ケタの 10 進数で、入出力文のファイル識別番号と対応する。

(2.2) 標準ラベル (SL 形式) のボリュームの場合、ファイル名と共にファイル順序番号を指定してもよい。もし、ファイル順序番号で指示した位置に指定した名前のファイルが存在しなければ、プログラムは強制終了となる。

(2.3) SL 形式の場合、ファイル順序番号はボリューム上のファイルの順番を意味する。ラベル無し (NL 形式) の場合には、ボリューム上の先頭のテープマークから使用するファイルの直後にあるテープマークまでのテープマーク数を意味する。

(2.4) DISP=CONT 指定は割り当てられているファイルをジョブ終了まで、以後のジョブステップに引き継ぐ指定である。指定すると、後続のジョブステップではそのファイルに関するファイル定義文を一切定義しなくても、同じアクセス名によってそのファイルへの読み書きが可能である。また、引き継ぎは直後のジョブステップのみならず、後続のジョブステップにも引き継がれる。ただし、後続のジョブステップで同じアクセス名をもつ別のファイルが定義されていると、そのジョブステップ内では参照できない。しかし、引き継ぎは有効であり、以後のジョブステップで参照可能である。

(2.5) 個人ファイルと同様に、磁気テープボリューム上のファイルもファイル名とボリューム通番の対応関係をカタログ登録簿に登録することができる。ただし、カタログ登録を行う磁気テープファイル名は個人ファイル名と同じ形式のファイル名とし、他人が使用するファイル名との一致を避けなければならない。カタログ登録されたファイルを参照する場合には、ボリューム通番の指定を省略することができる。

(2.6) SL 形式の磁気テープボリュームを使用する場合、ボリューム通番の代りに不特定ボリューム (SLMT) の指定が可能である。この場合には、多ボリュームであっても、VOL=SLMT とのみ記述する。

(2.7) 多ボリューム処理の場合でも、先頭のボリュームだけボリューム通番を指定すれば、システムは実行時に EOVS (end of volume) を検出して、

次のボリュームの取付要求をユーザに出すから、処理続行が可能である。

(2.8) ボリューム通番の数は 255 個まで記述可能である。

(2.9) システムからのマウント要求はジョブステップ実行開始前に上がる。ただし、AVR (automatic volume recognition) 登録の場合、最初のボリュームに対してマウント要求が上がらず、2 番目以降のボリュームに対して前のボリュームの処理が終了した時点で上がる。

(2.10) レコード形式、ブロック長、レコード長の指定方法について。

SL 形式の MT の場合、ファイル参照時および既存のファイルへの追加作成 (AD) 時には、ファイルラベルの情報が有効であるので、これらのパラメータは指定しない。もし、指定した場合、ファイルラベルの情報より制御文での指定の方が優先される。従って、指定に誤りがあると、以後正確にデータを読むことができなくなる。ファイル作成時には、任意の値を指定してよい。なお、ファイル作成時に、指定を省略すると、FORTRAN コンパイラの標準値でファイルが作成される。

NL 形式の MT の場合には、参照時、出力時ともにこれらのパラメータは省略することができない。

(3) 使用例

(3.1) 既存のファイルを参照する。

```

¥ USMT U01, FILE=H01. DATA1,
    VOL=NS0010, DISP=CAT (i)
¥ USMT U02, FILE=H01. DATA2 (ii)
¥ USMT U03, FILE=(/, 3),
    VOL=NM0005 (iii)
¥ USMT U04, FILE=DATA3,
    VOL=(NL0001, NL0002) (iv)
¥ USMT U11, FILE=(/, NL, 2),
    BSIZE=1200, RSIZE=120,
    RECFM=FB (v)
¥ USMT U12, FILE=H01. DATA4,
    DISP=UNCAT (vi)
¥ USMT U13, FILE=DATA5,
    VOL=SLMT (vii)

```

(i) 指定ボリューム通番のファイルを参照し、ジョブステップ終了後、カタログ登録簿に登録する。

(ii) ファイルの参照であるが、ボリューム通番はカタログ登録簿を参照するので省略する。

(iii) 指定ボリュームの先頭から3番目のファイルを参照する。

(iv) 多ボリュームの磁気テープを使用する。

(v) NL形式のテープのファイルを参照する。

(vi) 指定ファイルを参照し、ジョブステップ終了後、ファイル名とボリューム通番の対応関係をカタログ登録簿から削除する。

(vii) 不特定ボリュームを使用する。

(3.2) 新規ファイルを作成する。

```
¥ USMT U01, FILE=H01. DATA10,
  VOL=NM0030, DISP=CAT (i)
```

```
¥ USMT U02, FILE=(DATA20,
  OT, 2), BSIZE=1800,
  RSIZE=180, RECFM=FB,
  VOL=NL1000 (ii)
```

```
¥ USMT U03, FILE=(H01. DATA30,
  AD) (iii)
```

```
¥ USMT U04, FILE=(DATA40,
  FA), VOL=NM0030 (iv)
```

(i) 指定ボリュームにファイルを新規作成する。ファイルはボリュームの先頭から作成する。ジョブステップ終了後カタログ登録を行う。BSIZE, RSIZE, RECFM パラメータを省略しているから、ファイルはFORTRANコンパイラの標準値で作成される。

(ii) 指定ボリュームの2番目にファイルを新規作成する。なお、指定ファイルが既存のファイルであると、その内容を破壊して作成される。後のファイルも破壊するので注意を要する。

(iii) 既存ファイルにデータを追加する。既存データの最終位置からデータが追加される。最後のファイル以外にデータを追加すると後続ファイルの内容が破壊されるので注意を要する。

(iv) ファイルを追加作成する。ボリューム上に記録されている最終ファイルの後の位置に、磁気テープヘッドが位置付けされ、その位置から新規ファイルが作成される。

5.3.10 USDK文

(1) 記述形式

表 5.10

(2) 機能上の特記事項

(2.1) 個人ファイルは全て作成時に自動的にカタログ登録され、消去時に自動的にカタログ登録簿から削除される。

(2.2) ファイルの消去は作成者以外は不可能である。

(2.3) カatalog連結ファイルが使用可能である。カatalog連結ファイルとは、複数のファイルをグループ化して、カatalog登録簿上に登録したファイルである(使用例参照)。カatalog連結ファイルとして取り扱えるための条件は、次のとおりである。

(i) 順編成ファイルどおし、又は区分編成ファイルどおしであること。

(ii) 入力ファイルであること。

(iii) ファイルのレコード形式が同じであること。

(iv) ファイルのレコード長が同じであること。

(2.4) 追加領域が確保される場合、ファイルの領域が分割される個数(エクステント個数^注)は1ボリュームあたり8個までである。また、ファイルの増分は、そのファイルが他のジョブで使用されている場合には行われない。

(2.5) BSIZE, RSIZE, RECFMパラメータの指定方法について。

ファイル参照時および既存のファイルへの追加作成時にはこれらのパラメータを指定しない。指定した場合、もしファイル作成時の指定と異なると、以後正確にデータを読むことができなくなる。ファイル作成時、これらのパラメータを省略すると、FORTRANコンパイラの標準値でファイルが作成される。

(2.6) 区分編成ファイルに割り当てられている領域の中から1トラックがディレクトリ領域として使用され、登録可能なメンバの数はおよそ200個である。

(2.7) GO文の出力結果を、F4システムの個

注) MONITOR VIIシステムでのエクステントとは意味が異なる。MONITOR VIIでは増分回数を意味している。

人ファイルにHICS⁽¹⁰⁾でファイル伝送することを予定している場合には、本文を使用して個人ファイルに格納しておく。その際、RSIZE、BSIZEとRECFMは送り先データセットの値に合せておかねばならない。但し、RECFMではUを指定してはならない。

(3) 使用例

(3.1) 既存のファイルを参照する。

¥ USDK U01, FILE=H01.D1, DISP=DLT	(i)
¥ USDK U02, FILE=H01.D2, MEM=M1, PSW=P005	(ii)
¥ USDK U03, FILE=H01.D3, DISP=CONT	(iii)
¥ USDK U04, FILE=(H01.A.*)	(iv)

(i) アクセス名U01でファイルを参照し、ジョブステップ終了後、ファイルを消去する。

(ii) パスワードREAD/WRITE保護されている区分編成ファイルのメンバをアクセス名U02で参照する。

(iii) アクセス名U03でファイルを参照し、以降のジョブステップに引き継ぐ。

(iv) カタログ連結ファイルを使用する。例えば、カタログ登録簿に、H01.A.A, H01.A.B, H01.A.Cというファイルが登録されているとすると、これらを全て連結して参照する。

(3.2) ファイルを作成する。

¥ USDK U01, FILE=H01.DATA, TRK=(30,3,RLSE)	(i)
¥ USDK U02, FILE=(H01.A, OT), PSW=(PS10,WRITE), TRK=10	(ii)
¥ USDK U03, FILE=(H01.B, AD), MEM=M1, TRK=(/,5)	(iii)
¥ USDK U04, FILE=(H01.C, AD), MEM=(MM1, R)	(iv)
¥ USDK U07, FILE=(H01.D, AD), DISP=CONT	(v)

¥ USDK U08, FILE=H01. DATA30, BLK=(1000*20,5), PSW=P300	(vi)
--	------

¥ USDK U09, FILE=(H01.F, OT)	(vii)
------------------------------	-------

(i) 新たにファイルを作成する。領域の確保は一次量が30トラックで、容量が不足すると、3トラックずつ領域を追加する。ジョブステップ終了後、ファイルの未使用領域を解放する。なお、同一ファイル名が存在すると、エラーとなる。

(ii) 新たにファイルを作成し、それをパスワードWRITE保護する。もし同一のファイル名が存在すると、前のファイルの内容を破壊して新しいデータを書き込む。

(iii) 既存の区分編成ファイルにメンバを追加する。もし同一のメンバ名が存在するとエラーとなる。また、レコード形式およびレコード長が異なるメンバは作成できない。さらに増分値を指定値に変更する。

(iv) 既存の区分編成ファイルのメンバを置き換える。

(v) 既存の順編成ファイルにデータを追加する。既存データの最終位置からデータの追加が行われる。また、以降のジョブステップに引き継ぐ。

(vi) 新たにファイルを作成する。領域の確保は一次量は20,000バイト(1,000バイト×20ブロック)をトラックに換算した量で、不足の場合は5,000バイト(1,000バイト×5ブロック)をトラックに換算した量で追加確保する。このファイルをパスワードREAD/WRITE保護ファイルとする。

(vii) 既存のファイルを壊して新しいデータを出力する。

5.3.11 USWK文

(1) 記述形式

表 5.11

(2) 機能上の特記事項

(2.1) ジョブ処理中にUSWK文で作成されたファイルは後続ジョブステップに引きつがれ、ジョブ終了と共に消去される。

(2.2) 確保可能な最大領域の大きさは、ワークファイルとして定義されている作業域の全容量であ

る。ただし、空領域が要求量に満たない場合、ジョブは強制終了される。

(2.3) TRKまたはBLKパラメータは省略できない。

(2.4) FORTRANコンパイラの標準値を使用する場合、BSIZE, RSIZE, RECFMパラメータは省略する。

(3) 使用例

```

¥ GO EF=(H01.A,P400),
    EMEM=P1                                ①
¥ USWK U01,TRK=(30,2)
¥ USWK U02,BLK=(1000*30,5)
¥ GO EF=(H01.B,B500),
    EMEM=P2                                ②
¥ USDK U02,FILE=H01.DATA3,
    PSW=3000
¥ GO EF=(H01.C,C200),
    EMEM=P3                                ③
  
```

(i) ①のジョブステップではアクセス名U01とU02のワークファイルが使用できる。

(ii) ②のジョブステップではアクセス名U01のワークファイルとアクセス名U02の個人ファイルが使用できる。

(iii) ①のジョブステップと同様に、③のジョブステップではアクセス名U01とU02のワークファイルが使用できる。

5.3.12 CF文

(1) 記述形式

表 5.12

(2) 機能上の特記事項

(2.1) CF文にはその直前に親の制御文が必要である。USDK文、USMT文およびFD文はCF文の親の制御文となりえる。また、GOオプション指定のGO文、LINKGO文に続けて入力するデータカードに連結する場合、親の制御文はこれら各マクロ内に存在するから、ユーザは親の制御文を意識しなくてもよい。

(2.2) 連結ファイルの順序は、連結するファイルを定義する制御文の順序通りである。

(2.3) 連結しえるファイルには次の制約がある。

(i) 入力ファイルであること。

(ii) 各ファイルのレコード形式が同じであること。

(iii) 各ファイルのレコード長が同じであること。

(iv) 各ファイルのブロック長が等しいか、または直前のファイルよりも小さいこと。

(v) 入力装置がDK, MT, CRであること。

(vi) 連結するファイルの組合せが順編成ファイル-順編成ファイル、順編成ファイル-区分編成ファイルのメンバ、区分編成ファイルのメンバ-区分編成ファイルのメンバのいずれかであること。

(2.4) 入力装置がMTの場合には同一の装置が割りられる。また、VOLパラメータを省略するとカタログ登録簿が参照される。

(3) 使用例

(3.1) MTファイルに個人ファイル(区分編成ファイルのメンバ)を連結する。即ち、アクセス名U01でMTファイルZ3, 個人ファイルH01.Z1のメンバP1およびファイルH01.Z2のメンバM1, M2を連結して読み込む。

```

¥ USMT U01,FILE=Z3,
    VOL=NM4000
¥ CF DK,FILE=H01.Z1, MEM=P1,
    PSW=PS10
¥ CF DK,FILE=H01.Z2,
    MEM=(M1,M2)
  
```

(3.2) 個人ファイル(順編成ファイル), MT, カードファイルを連結する。

```

¥ USDK U03,FILE=H01.A10
¥ CF MT,FILE=(/,3), VOL=SLMT
¥ CF CR
  入力データカード
¥ CF DK,FILE=H01.A3,
    PSW=PS10
¥ CF DK,FILE=H01.A2,
    PSW=PS10
¥ CF MT,FILE=H01.D1
  
```

(3.3) データカードに個人ファイル, MTファイルを連結する。即ち、READ(5,...)により、カードファイル, 個人ファイル, MTファイルの順に

連結して読み込む。

¥ GO <div>入力データカード</div> ¥ CF DK, FILE=H01.A, MEM=D1, PSW=PS10 ¥ CF MT, FILE=D2, VOL=NS0001
¥ FORTC PARAM=GO, SF=(H01.A1, PS10) <div>入力データカード</div> ¥ CF DK, FILE=H01.B, MEM=(M1, M2) ¥ CF CR <div>入力データカード</div>

(注) 親の制御文(¥FD UIN=+)はGO文, FORTC文の中で展開されているので指定する必要がない。ファイル識別番号の5はアクセス名UINとなる。

5.3.13 SAME 文

(1) 記述形式

表 5.13

(2) 機能上の特記事項

(2.1) 本マクロは一つのジョブステップで複数のMTファイルを作成または参照する場合に有効である。

(2.2) SAME文の親の制御文はUSMT文, 又はFD文(ただし, 磁気テープ装置使用)でなければならない。本マクロによって共用される装置は親の制御文が使用する装置である。

(2.3) 親の制御文でNL形式の磁気テープを指定している時は, SAME文でもNL形式の指定でなければならない。また, ボリュームは親の制御文で使用するボリュームと異なるボリュームは使用できない。

(2.4) SAME文でボリューム通番を省略した場合, およびVOL=SLMTを指定した場合, 指定ファイルは親の制御文のボリュームの中から探し出される。

(2.5) SAME文で指定したファイルが同時に処理されることはない。

(2.6) カタログ登録簿からボリューム通番を参

照する場合には, REFサブパラメータの指定が必要である。

(3) 使用例

¥ USMT U11, FILE=B, VOL=NM0001 ¥ SAME U12, FILE=C ¥ SAME U13, FILE=H01.A, DISP=REF ¥ SAME U14, FILE=E, VOL=NL0002 ¥ SAME U15, FILE=H01.D, VOL=NS0001, DISP=(CAT, REF)	(i) (ii) (iii) (iv) (v)
---	-------------------------------------

(i) ボリューム通番「NM0001」の中からファイルBを参照する。

(ii) ボリューム通番の指定がないから, ファイルCはボリューム「NM0001」の中から探し出される。

(iii) DISP=REFの指定があるから, ファイルH01.Aはカタログ登録簿のボリューム通番の中から探し出される。

(iv) ファイルEはボリューム通番「NL0002」の中から探し出される。

(v) ファイルH01.Dをいったんカタログ登録簿のボリューム通番の中から探し出し, ジョブステップ終了時にカタログ登録簿のボリューム通番を「NS0001」に変更する。

5.3.14 XY 文

(1) 記述形式

表 5.14

(2) 機能上の特記事項

(2.1) 本文はマクロ展開形の中で, アクセス名U99(または, FT99F001)を使用する。XY出力を行う場合にはユーザプログラムの中のPLOTSサブルーチンの引数で99を指定しなければならない。

(2.2) 本文は1ジョブ当たり1枚しか定義できない。複数ジョブステップで定義する場合はジョブを複数に分けなければならない。

(3) 使用例

(3.1) XYプロッタ出力を指示し, かつ, コンパイラオプションとともに, ロードオプションを指

定する。

```
¥ FORTC PARAM=(GO/MAP),
    SF=H01.A
¥ XY
```

(3.2) XYプロッタ出力を必要としない。

```
¥ GO EF=(H01.B, PS20), EMEM=M1
¥ XY DEBUG
```

5.3.15 UPDATE 文

(1) 記述形式

表 5.15

(2) 機能上の特記事項

(2.1) 挿入レコード、置き換えレコード、マージデータ、更新ファイルは全てカードイメージであり、かつ、レコード長は 80 バイト、ブロック長は 1040 バイト、レコード形式は固定長でなければならない。

(2.2) ファイルの更新は補助制御文による方法とマージデータによる方法とがある。

(i) 補助制御文による方法

- INSERT 文でレコードを挿入する。
- REPLACE 文でレコードを置き換える。
- DELETE 文でレコードを削除する。
- RENUM 文で識別順序を付け直す。

(ii) マージデータによる方法

- 識別順序を使用してレコードの挿入、レコードの置き換えを行う。

(2.3) 識別順序を付け直ししながら、INSERT 文、REPLACE 文、DELETE 文、マージデータによる更新が可能である。このとき、RENUM 文は他の更新データよりも前に置かなければならない。

(2.4) 更新したレコードを磁気テープに出力する場合は、常にボリュームの先頭から書き込むので注意を要する。

(2.5) 挿入するレコードに識別順序を付ける指定をした場合に、もし挿入データ数が番号付け可能数を超えたならば、超えた追加データには番号が付けられない。

(2.6) LIST オプションを指定しないと印刷出力は行われない。ただし、次の情報は自動的に出力

される。

- (i) 補助制御文のリスト
- (ii) 削除されたレコードの内容
- (iii) 挿入されたレコードと、そのモジュール内相対番号
- (iv) 置き換えられたレコードの内容と置き換えたレコードの内容及びモジュール内相対番号
- (v) 出力モジュールの新しいレベル番号、ブロック数
- (vi) REP パラメータによってメンバを置き換えた情報

ただし、NOMSG オプションを指定すると上記の情報は出力されない。

(2.7) 本マクロでは標準形式のファイルしか処理できない。会話型処理で作成した自由形式のファイルは、CONVERT コマンドで標準形式に変換しなければ処理できない。

(2.8) 区分編成ファイルの場合、PARAM パラメータでオプションを指定しないと、自動的に指定メンバの内容は更新後のデータに置き換えられ、NEWNAME オプションで新メンバ名を指定すると、更新前のメンバをそのままにして更新後のデータを新メンバとして登録する。ただし、他のオプションを指定する場合には、必ず REP または NEWNAME オプションと共に指定しないと更新後のデータを残せない。また、順編成ファイルの場合には、REP および NEWNAME オプションは機能せず、指定ファイルの内容は必ず更新後のデータに書き換えられる。

(2.9) 識別順序は識別記号と順序番号からなる。システムの省略値は、識別欄は 73 から 75 桁であり、順序欄は 76 から 80 桁である。挿入処理は識別順序をキーとして行われる。^(註)

(2.10) 更新ファイルのファイル編成を更新後に変更することはできない。

(3) 使用例

指定のメンバのレコードに、INSERT 文でデータを挿入し、REPLACE 文でデータを置き換え、

^(註) 但し、会話型処理では識別記号は数字でなければならない。

DELETE 文でレコードを削除し、マージデータを混ぜ合せ、識別順序を付け直し、更新したレコードを元のメンバの内容と置き換え、更新したデータの内容を出力する。

```

¥ UPDATE PARAM=(REP, LIST),
  OF=(H01.A1, PS10), MEM=M1
/ RENUM
/ INSERT AAA00200
  挿入データカード
/ REPLACE AAA00500: AAA00800
  置き換えデータカード
/ DELETE AAA01500: BOT
  マージデータカード

```

5.3.16 INSERT 文

個人ファイルにレコードの挿入を指示する文である。本文は UPDATE 文の挿入機能を独立させたものである。

(1) 記述形式

表 5.16

(2) 機能上の特記事項

(2.1) UPDATE 文の特記事項を参照のこと。

(2.2) 更新するファイルに識別順序がないと、レコードを挿入できる位置は更新レコードの先頭、または最後であり、途中には挿入できない。途中挿入が必要ならば、識別順序を付加しておかなければならない。

(2.3) INIT, INCR パラメータを省略するとシステムで定められた識別順序番号が付けられる。

(3) 使用例

```

¥ INSERT IDSEQ=TOP,
  OF=(H01.A2, PS20), MEM=M1
  挿入データカード
¥ INSERT IDSEQ=AAA00100,
  OF=(H01.B, PS02),
  INIT=AAA00110, INCR=10 (ii)
  挿入データカード
¥ INSERT PARAM=(NEWNAME=M3,
  LIST), IDSEQ=BOT,

```

OF=(H01.C, PS03), MEM=M2
(iii)

挿入データカード

(i) 指定メンバのレコードの先頭にカード入力のデータを挿入し、元のメンバの内容と置き換える。

(ii) 指定の順編成ファイルに対して、指定の識別順序をもつレコードの後にデータを挿入し、かつ挿入レコードに識別順序を付加する。指定の識別順序のレコードが存在しないとエラーとなる。

(iii) 指定メンバのレコードの最後にデータを挿入し、旧メンバは元のままに残し、同一ファイル内に更新したレコードに新メンバ名を付加して保存する。また、新しいメンバの内容を出力する。

5.3.17 MERGE 文

マージデータを個人ファイルの中に混ぜ合せることを指示する文である。本文は UPDATE 文の機能の一部を独立させたものである。

(1) 記述形式

表 5.17

(2) 機能上の特記事項

(2.1) UPDATE 文の特記事項を参照のこと。

(2.2) マージデータには、更新するファイルのレコードと同じ識別順序欄の位置に識別順序を指定しなければならない。

(2.3) 更新するファイルのレコードは、同一識別記号グループ内で順序番号の昇順にレコードが並んでいなければならない。昇順でないレコードは識別順序の付け直しを行っておく必要がある。

(2.4) マージデータを複数与えるときには、同一識別記号グループ内で昇順に並んでいなければならない。

(2.5) 更新するファイルのレコードとマージデータの識別順序が一致すると置き換えとなる。マージデータの識別記号と一致するレコードはあるが、同一識別記号グループ内での順序番号が一致しない場合には、順序番号が昇順に並ぶ位置に挿入される。

(2.6) マージデータの識別記号と一致するレコードがない場合、エラーとなる。

(2.7) 更新するファイルの識別記号がない場合には、全てのレコードは同一識別記号グループとみ

なして処理される。

(3) 使用例

```
¥ MERGE PARAM=(REP,LIST),
  OF=(H01.A3,P003),MEM=M1
```

(i)

マージデータカード

```
¥ MERGE OF=H01.B
```

(ii)

マージデータカード

```
¥ MERGE PARAM=(NEWNAME=M3,
  CHKLVL=5),OF=(H01.C,
  PS02),MEM=M2
```

(iii)

マージデータカード

(i) 指定メンバのレコードにマージデータを混ぜ合せ、元のメンバの内容と置き換え、更新したレコードの内容を出力する。

(ii) 指定の順編成ファイルのレコードにマージデータを混ぜ合せ、更新したレコードに置き換える。

(iii) 指定ファイルのレベル番号（ソースプログラムの更新回数を示す値で、ディレクトリエントリに含まれる制御情報）が指定値と等しくないと更新処理を行わない。更新処理を行ったときには更新前のメンバをそのまま残して、更新したデータを新しいメンバとして追加する。

5.3.18 OMIT文

個人ファイル内のレコードを削除することを指示する文である。本文はUPDATE文の削除機能を独立させたものである。

(1) 記述形式

表 5.18

(2) 機能上の特記事項

(2.1) UPDATE文の特記事項を参照のこと。

(2.2) ファイル内の指定レコードを削除したり、複数の連続するレコードを削除することができる。

(3) 使用例

```
¥ OMIT PARAM=(REP,LIST),
  IDSEQ=AAA00100:AAA00200,
  OF=(H01.A,PS01),MEM=M1
```

(i)

```
¥ OMIT IDSEQ=TOP:AAA00100,
```

```
OF=H01.B
```

(ii)

```
¥ OMIT PARAM=(NEWNAME=M3),
  IDSEQ=AAA00500:BOT,
  OF=(H01.C,PS02),MEM=M2
```

(iii)

(i) 指定メンバの識別順序AAA00100から識別順序AAA00200までのレコードを削除してメンバの内容を置き換え、新しいメンバの内容を出力する。

(ii) 指定の順編成ファイルの先頭のレコードから指定の識別順序までのレコードを削除し、ファイルの内容を置き換える。

(iii) 指定メンバの指定の識別順序から最終のレコードまでを削除し、元のメンバはそのままにして、同一ファイル内に指定のメンバ名を追加して保存する。

5.3.19 RENUM文

個人ファイルの識別順序の付け直しを指示する文である。本文はUPDATE文の機能の一部を独立させたものである。

(1) 記述形式

表 5.19

(2) 機能上の特記事項

(2.1) UPDATE文の特記事項を参照のこと。

(2.2) IDSEQパラメータを省略すると全レコードの識別順序の付け直しとなる。このとき、INITパラメータも省略すると、同一識別記号グループごとに順序番号のみがINCRの増分値（省略のときは10）を初期値として付け直される。

(3) 使用例

```
¥ RENUM OF=(H01.A3,P003),
  MEM=M1
```

(i)

```
¥ RENUM PARAM=LIST,
  IDSEQ=TOP:AAA15000,
  OF=H01.B2,INIT=AAA00000,
  INCR=30
```

(ii)

```
¥ RENUM PARAM=(NEWNAME=M3),
  IDSEQ=BBB00500:BBB01000,
  OF=H01.C5, MEM=M2,
  INIT=CCC00000, INCR=20
```

(iii)

(i) 指定メンバの全レコードに、同一識別記号グループごとに順序番号のみ付け直し、メンバを更新した内容に置き換える。

(ii) 指定の順編成ファイルのレコードを先頭から指定の識別順序までのレコードを初期値と増分値の指定に従って付け直し、更新したファイルの内容を出力する。ファイルの内容は更新したものに置き換える。

(iii) 指定メンバのレコードを指定の識別順序から識別順序までのレコードに指定の初期値と増分値で識別順序を付け直し、旧メンバはそのまま保存し、更新したレコードに新しいメンバ名をつけて同一ファイル内に追加作成する。

5.3.20 DELETE 文

(1) 記述形式

表 5.20

(2) 機能上の特記事項

(2.1) ファイルが消去されるときには、カタログ登録簿からも削除される。

(2.2) 区分編成ファイルの場合、ファイル名だけ指定するとファイル全体が消去される。メンバ名を指定すると、指定のメンバだけが消去され、ファイル自体は残る。このとき、消去したメンバが使用していた領域はコンデンスしなければ使用可能領域とはならない。コンデンスは会話型処理で行う。

(2.3) ファイルの消去は作成者以外不可能である。

(3) 使用例

(3.1) 順編成ファイルまたは区分編成ファイルを消去する。

```
¥ DELETE OF=H01.MAIN
```

(3.2) 区分編成ファイルのメンバを消去する。
(ファイルは残す。)

```
¥ DELETE OF=(H01.B, PS30),  
MEM=(M1, M2, M3)
```

5.3.21 CRTODK 文

(1) 記述形式

表 5.21

(2) 機能上の特記事項

(2.1) 入力データの内容はソースモジュールでなければならない。出力ファイルのブロック長は 1040 バイト、レコード長は 80 バイト、レコード形式は固定長となる。なお、NF パラメータは単ファイルの指定しかできない。

(2.2) 区分編成ファイルにメンバを登録する場合(複数メンバも可)には、メンバ単位に補助制御文(ADD 文)を先頭に付加しなければならない。ファイルを順編成とする場合には、補助制御文を付加してはならない。

(2.3) オプションパラメータは、ファイル編成を区分編成とする場合は ADD 文の中で指定し、順編成とするときには PARAM パラメータで指定する。

(2.4) 既存の区分編成ファイルに作成する場合、ファイル内に同名のメンバがあるとエラーとなる。このため、メンバを置き換えるためには REP オプションを指定しなければならない。

(2.5) LIST オプションを指定しない場合でも次の情報が自動的に出力される。

(i) 補助制御文リスト

(ii) 登録したデータのカード枚数

(iii) 登録したデータのブロック数

(iv) REP オプションの指示により置き換えたメンバの情報

ただし、NOMSG オプションを指定すると上記の情報は出力されない。

(2.6) 入力データとして、1 桁目が ¥ と / で始まるカードを含めてはいけない。

(2.7) 新規作成ファイルの場合には TRK パラメータは省略できない。

(2.8) 特に指定がない場合、出力は追加(AD)処理となる。

(3) 使用例

(3.1) 新たに区分編成ファイルを作成して複数のメンバを登録し、指定メンバのみ登録カードの内容を印刷出力する。さらにファイルをパスワード WRITE 保護ファイルとする。

```

¥ CRTODK NF=H01.A,
  PSW=(P200,WRITE),
  TRK=(10,2,RLSE)

```

```

/ ADD M1,LIST

```

登録カード

```

/ ADD M2

```

登録カード

```

/ ADD M3

```

登録カード

(3.2) 新たに順編成ファイルとしてカードの内容を登録し、その内容をリストに出力する。

```

¥ CRTODK NF=H01.C,ORG=PS,
  PARAM=LIST,TRK=50

```

登録カード

(3.3) カードの内容を既存の区分編成ファイルの同名のメンバと置き換える。

```

¥ CRTODK NF=H01.D
/ ADD M5,REP

```

登録カード

5.3.22 CRTOMT文

(1) 記述形式

表 5.22

(2) 機能上の特記事項

(2.1) CRTODK 文の特記事項 (2.1), (2.5), (2.6) を参照のこと。

(2.2) 磁気テープは SL 形式のものに限る。

(2.3) NF パラメータを省略し、補助制御文 (ADD 文) を付加すると多ファイルの登録が可能である。

(2.4) 多ファイルの作成は常に出力磁気テープの先頭から行う。

(2.5) NV を省略すると SLMT の指定となる。

(2.6) 特に指定がない場合、出力は新規作成 (OT) 処理となる。

(3) 使用例

(3.1) カードの内容を指定のボリュームに新規ファイルとして登録する。このとき、ボリューム NM0005 に出力しきれない場合にはボリューム

NM0006 に出力する。

```

¥ CRTOMT NF=MTFILE,
  NV=(NM0005,NM0006)

```

登録カード

(3.2) ファイルを追加作成する。指定ボリューム上に記録されている最終ファイルの後の位置から新規ファイルが追加作成される。

```

¥ CRTOMT NF=(MTDATA,FA),
  NV=NL0001

```

登録カード

(3.3) 不特定ボリュームを指定し、既存のファイルにカードの内容を追加し、カードの内容を印刷出力する。

```

¥ CRTOMT PARAM=LIST,
  NF=(DATA10,AD)

```

登録カード

(3.4) 指定ファイルが既存ならその先頭からカードの内容を出力する。このため、指定ファイルの内容は破壊され、このファイルの後のファイルも破壊される。指定ファイルが存在しなければ、新規ファイルが作成される。

```

¥ CRTOMT NF=(FILE,OT),
  NV=NS0001

```

登録カード

5.3.23 CRTOLP 文

(1) 記述形式

表 5.23

(2) 機能上の特記事項

入力カードの中に 1 桁目が ¥ で始まるカードと、1~2 桁目が // で始まるカードは含めることができない。

(3) 使用例

カードの内容をリストに出力する。

```

¥ CRTOLP

```

入力カードデータ

5.3.24 DKTODK 文

(1) 記述形式

表 5.24

(2) 機能上の特記事項

(2.1) 出力ファイルを入力ファイルと異なる編成とする場合には ORG パラメータを指定しなければならない。ただし、ファイルの内容がロードモジュールの場合には、入力、出力とも区分編成でなければならない。

(2.2) ORG=PS を指定すれば、区分編成ファイルの複数メンバを 1 つの順編成ファイルとして結合して複写することが可能である。

(2.3) 入力ファイルが順編成で出力ファイルが区分編成の場合、メンバ名を NEWNAME オプションで指定する。

(2.4) 入力ファイルが区分編成ファイルで、メンバを複数指定しない場合には、NEWNAME オプションで新しいメンバ名を指定することができる。複数メンバを指定する場合には入力メンバと同じメンバ名で出力される。

(2.5) 既存の区分編成ファイルに追加出力する場合、同じメンバ名のものと置き換えるには REP オプションを指定する。

(2.6) 新規ファイルに出力する場合には必ず TRK パラメータを指定しなければならない。

(2.7) 特に指定がない場合、出力は追加 (AD) 処理となる。

(3) 使用例

¥ DKTODK OF=(H01.A,P100),
NF=H01.B (i)

¥ DKTODK OF=H01.C,NF=H01.D,
MEM=(M1,M2) (ii)

¥ DKTODK OF=(H01.E,P030),
NF=H01.F, MEM=(M1,M2,M3),
TRK=(10,2),ORG=PS,
PSW=(P300,WRITE) (iii)

¥ DKTODK OF=(H01.F,P300),
PARAM=(NEWNAME=PROC3),
NF=H01.G,ORG=PO,TRK=20 (iv)

(i) 指定の順編成ファイルを既存の順編成ファイルに追加する。AD サブパラメータは省略できる。

(ii) メンバを既存の区分編成ファイルに追加する。出力されたメンバ名は入力したメンバ名と同じである。

(iii) メンバを結合して新規順編成ファイルとして作成し、パスワードWRITE 保護ファイルとする。

(iv) 順編成ファイルを区分編成ファイルとして新規に作成する。このときのメンバ名は NEWNAME オプションで指定する。

5.3.25 DKTOMT 文

(1) 記述形式

表 5.25

(2) 機能上の特記事項

(2.1) 本文ではロードモジュールの複写はできない。その必要がある場合には BACKUP 文 (第 5.3.31 項参照) を使用すること。

(2.2) 個人ファイルが順編成ファイルの場合、OF, OF1, OF2, OF3, OF4 の順に多ファイルの指定ができる。このとき、出力ファイルはそれぞれ NF, NF1, NF2, NF3, NF4 と指定しなければならない。ただし、区分編成ファイルは単ファイルしか入力できない。

(2.3) 区分編成ファイルの複数のメンバを連結し 1 つの順編成ファイルとして出力したい場合には NF で一つのファイル名を指定する。NF パラメータの指定がないと、メンバ毎に順編成ファイルとなり、ファイル名は入力したメンバ名と同名のものがつけられる。

(2.4) NV パラメータを省略した場合は SLMT の指定となる。

(2.5) 出力を多ファイル指定した場合、NV パラメータは単ボリュームまたは SLMT の指定でなければならない。単ファイル出力の場合、多ボリュームの指定は可能である。

(2.6) 磁気テープは SL 形式に限る。

(2.7) 特に指定がない場合、出力は新規作成 (OT) 処理となる。

(3) 使用例

```

¥ DKTOMT OF=(H01.A, PS30) (i)
¥ DKTOMT OF=H01.B,
  MEM=(M1,M2,M3) (ii)
¥ DKTOMT OF=(H01.C,C500),
  MEM=+, NF=(MTFILE,FA),
  NV=(NL0010,NL0020) (iii)
¥ DKTOMT OF=H01.D,
  OF1=(H01.E,E800),
  OF2=(H01.F,F800),
  OF3=(H01.G,F230),
  OF4=H01.H, NF=MT10,
  NF1=MT20,NF2=MT30,
  NF3=MT40,NF4=MT50,
  NV=NL0020 (iv)

```

(i) 順編成の個人ファイルを不特定ボリューム指定で磁気テープの先頭から出力する。マウント要求は実行時に上がる。

(ii) 区分編成ファイルの指定メンバを多ファイルとして出力する。

(iii) 区分編成ファイルの全メンバを1つのファイルに結合して、磁気テープ上の最終ファイルの後から出力する。

(iv) 順編成ファイルを複数個磁気テープに複写する。このとき、NVパラメータは単ボリューム指定またはSLMT指定でなければならない。

5.3.26 DKTOLP文

(1) 記述形式

表 5.26

(2) 機能上の特記事項

(2.1) 出力可能な個人ファイルの内容は、ソースモジュールと一般データであり、ファイル編成は順編成または区分編成でなければならない。なお、ブロック長、レコード長、レコード形式は任意である。

(2.2) RECNOの指定を省略すると、指定のファイル、またはメンバの先頭レコードから最後のレコードの内容を印刷出力する。

(2.3) HEXAオプションを指定すると、レコードの内容(EBCDIC文字)をそのまま印刷するとともに、16進表現に変換した印刷もする。その際、

EBCDICコード表にないコードはピリオドとして印刷される。

(3) 使用例

```

¥ DKTOLP OF=H01.A,COPY=3,
  MEM=(M1,M2,M3) (i)
¥ DKTOLP OF=(H01.A2,PS30),
  RECNO=TOP:120 (ii)
¥ DKTOLP OF=(H01.A3,PS40),
  PARAM=HEXA, MEM=M4 (iii)

```

(i) 区分編成ファイルの指定メンバを3部出力する。

(ii) 指定の順編成ファイルの先頭レコードから指定の相対番号までを印刷出力する。

(iii) 区分編成ファイルの指定メンバを印刷形式をHEXAモードで印刷出力する。

5.3.27 MTTODK文

(1) 記述形式

表 5.27

(2) 機能上の特記事項

(2.1) 出力ファイルは順編成、区分編成共に可能である。ただし、後者の場合、ORG=POを指定しなければならない。

(2.2) 出力ファイルを順編成とすると、OF, OF1, OF2, OF3, OF4に対して、それぞれNF, NF1, NF2, NF3, NF4を指定すれば、多ファイルの指定が可能である。出力ファイルを区分編成とする場合には入力ファイルは単ファイルの指定しかできないから、必要な枚数だけMTTODK文を用意する。なお、多ファイルの複写の場合、ブロック長、レコード長、レコード形式が全ファイルとも同じでなければならない。

(2.3) 出力ファイルのブロック長、レコード長、レコード形式は入力ファイルのそれと同じになる。このため、既存の個人ファイルに追加出力する場合には注意を要する。

(2.4) 出力ファイルを区分編成とする場合、NEWNAMEオプションで新メンバ名を指定しなければならない。また、既存のファイルへ追加作成するときに、ファイル内に同名のメンバ名があるならばREPオプションを指定して置き換える。

(2.5) NL形式の磁気テープを使用する場合には、BSIZE, RSIZE, RECFMパラメータの指定を省略できない。

(2.6) 個人ファイルを新規作成する場合、TRKパラメータの指定を省略できない。

(2.7) 多ファイルを入力する場合、OVパラメータは単ボリュームまたは、SLMT指定でなければならない。なお、単ファイル入力の場合には、単ボリューム、多ボリューム、SLMTの指定が可能である。

(2.8) 多ファイルを入力し、順編成の多ファイルを作成するときにパスワードの指定をすると、全ファイルが同じパスワードの保護となる。

(2.9) OVパラメータを省略すると、カタログ登録簿が参照される。

(2.10) 特に指定がない場合、出力は追加(AD)処理となる。

(3) 使用例

```

¥ MTTODK OF=MF01,
  OV=NM0005,NF=H01.A,
  TRK=(10,3) (i)
¥ MTTODK PARAM=(NEWNAME=M1),
  OF=(/,3),NF=H01.B,
  TRK=(20,2),ORG=PO,OV=SLMT
¥ MTTODK PARAM=(REP,
  NEWNAME=M3),OF=H01.MT,
  NF=H01.DATA,ORG=PO,
  PSW=P003 (iii)
¥ MTTODK OF=(/,NL,4),
  NF=(H01.NLDATA,OT),
  BSIZE=2160,RSIZE=720,
  RECFM=FB (iv)
¥ MTTODK OF=MDAT1,
  OF1=MDAT2,OF2=MDAT3,
  OV=NM0005,NF=H01.MT1,
  NF1=H01.MT2,NF2=H01.MT3,
  TRK=(50,5),PSW=(7007,
  WRITE) (v)

```

(i) 個人ファイルを新規作成し、指定ボリューム上のファイルを順編成で複写する。

(ii) 不特定ボリュームの先頭から3番目のファイルを新規ファイルとして区分編成で作成する。

(iii) ボリューム通番はカタログ登録簿を参照して、指定ファイルを既存の区分編成ファイルに出力し、同名のメンバと置き換える。なお、追加出力を示すADサブパラメータは省略できる。

(iv) NL形式の磁気テープを指定し、既存の順編成ファイルの内容を壊して新たにファイルを作成する。

(v) 指定ボリューム上の多ファイルを新規に順編成ファイルとして出力する。なお、領域確保およびパスワードパラメータは全ファイルに共通である。

5.3.28 MTTOMT文

(1) 記述形式

表 5.28

(2) 機能上の特記事項

(2.1) 複写可能なファイルの内容はソースモジュール、オブジェクトモジュール、および一般データである。

(2.2) SL形式の磁気テープの場合には、単ファイルおよび多ファイルの複写が可能であるが、入力、出力ファイルともブロック長、レコード長、レコード形式を指定してはならない。

(2.3) NL形式の磁気テープの場合には、SL形式で出力され、単ファイルの複写しかできない。また、ブロック長、レコード長、レコード形式の指定を省略できない。NL形式からNL形式への複写とNL形式の多ファイルの取り扱いはMTCOPY文(第5.3.30項参照)で行える。

(2.4) OVパラメータを省略するとカタログ登録簿が参照される。また、NVを省略すると、SIMTの指定となる。

(2.5) 多ファイルの複写の場合には、常に磁気テープの先頭から出力する。また、多ファイルの複写のときには多ボリュームを指定できない。

(2.6) 特に指定がない場合、出力は新規作成(OT)処理となる。

(3) 使用例

```

¥ MTTOMT OFS=DATA, NFS=DATA1,
  OV=NM0010, NV=NM0020 (i)
¥ MTTOMT OFS=(/,NL,2),
  NFS=DATA3, BSIZE=514,
  RSIZE=514, RECFM=F,
  NV=NM0030 (ii)
¥ MTTOMT OFM=(TEST0, TEST1,
  TEST2), OV=NS0050 (iii)
¥ MTTOMT OFM=+, OV=SLMT (iv)
¥ MTTOMT OFM=(*:DATA5) (v)
¥ MTTOMT OFM=8, OV=SLMT (vi)

```

- (i) SL形式の磁気テープのファイルを複写する。
- (ii) NL形式の磁気テープのファイルをSL形式の磁気テープに複写する。
- (iii) SL形式の磁気テープの複数ファイルを不特定ボリュームに複写する。
- (iv) 不特定ボリューム指定の磁気テープの全ファイルを不特定ボリューム指定の出力ボリュームに複写する。
- (v) 入力ボリュームはカタログ登録簿を参照し、先頭から指定のファイルまでを全て複写する。
- (vi) SL形式の磁気テープの先頭から8番目までのファイルを複写する。

5.3.29 MTTOLP文

(1) 記述形式

表 5.29

(2) 機能上の特記事項

(2.1) 入力するMTファイルの内容は、ソースモジュール、一般データであり、ブロック長、レコード長、レコード形式は任意である。

(2.2) RECNOの指定を省略すると、指定ファイルの先頭レコードから最終レコードまでの内容を印刷出力する。

(2.3) OFで指定できるファイル名またはファイル順序番号は1つであり、多ファイルの指定はできない。

(2.4) NL形式の磁気テープの場合には、BSIZE, RSIZE, RECFMの指定を省略できない。

(2.5) OVの指定を省略するとカタログ登録簿が参照される。

(2.6) DKTOLP文の特記事項の(2.3)を参照のこと。

(3) 使用例

```

¥ MTTOLP COPY=3, OF=DATA5,
  OV=NS0030 (i)
¥ MTTOLP OF=(/,NL,5),
  RECNO=TOP:30, BSIZE=514,
  RSIZE=514, RECFM=F (ii)

```

(i) SL形式の磁気テープの指定ファイルの先頭から最終レコードまでを3部印刷出力する。

(ii) NL形式の磁気テープの指定ファイルの先頭から30番目までのレコードの内容を印刷出力する。

5.3.30 MTCOPY文

(1) 記述形式

表 5.30

(2) 機能上の特記事項

(2.1) 本マクロはSL, NLどちらの形式でも使用可能であるが、デッドコピーのときはボリューム通番も複写され入力ボリュームと同じ通番となる。ボリューム通番を異なるものにしたい場合には、PARAM=Hを指定する。

(2.2) 入力ボリュームが複数の場合には、先頭ボリュームに準じて、SL形式、またはNL形式で統一されていなければならない。

(2.3) PARAM=Aを指定し、出力磁気テープボリュームの既書き込み済みのファイルの後へ書き込む場合、出力磁気テープのラベル形式に合わせなければならない。

(2.4) 入力磁気テープは、SL形式の場合、最大4本まで可能であるが、NL形式の場合は1本しか入力できない。出力磁気テープは共に1本に限る。

(2.5) SL形式の複数ボリュームのファイルを統合して、単ボリュームに多ファイルを作成する場合、処理中に多ボリュームにわたるような事態になって処理を中止しても、処理中のファイル以外は正常である。なお複数ボリュームを入力する場合には、必ずボリューム通番をOV, OV1, OV2, OV3の順に指定しなければならない。

(2.6) 異なる記録密度(1600または800BPI)で出力できる。DENパラメータを省略すると、1600BPIで出力される。

(2.7) PARAMパラメータを省略するとデッドコピーと見なされる(即ち、PARAM=(*)の指定となる)。

(3) 使用例

¥ MTCOPY	(i)
¥ MTCOPY OF=(/,NL)	(ii)
¥ MTCOPY PARAM=H, OV=NM0010, OV1=NM0020, NV=NL0050	(iii)

- (i) SL形式の磁気テープをデッドコピーする。
- (ii) NL形式の磁気テープをデッドコピーする。
- (iii) 2本のSL形式の磁気テープを1本に統合する。

5.3.31 BACKUP文

(1) 記述形式

表 5.31

(2) 機能上の特記事項

(2.1) バックアップ形式は、DKボリューム上の区分編成ファイルに関する制御情報までもファイルといっしょに退避する特殊な形式である。このため、退避されたファイルはRESTORE文を使用してDKに復元しないと使用できない。

(2.2) BACKUP文は、DKTOMT文では処理できないロードモジュールのメンバの複写ができる。

(2.3) NVを省略するとSLMTの指定となる。

(2.4) 出力ファイル名を省略すると、入力ファイルと同じファイル名が付けられる。

(2.5) MEMパラメータを省略すると全メンバを退避する。

(3) 使用例

¥ BACKUP OF=(H01.A,PS01), MEM=M1,NV=NM0030	(i)
¥ BACKUP OF=H01.B,NF=MAIN	(ii)

- (i) 区分編成ファイルの指定メンバのみ退避し、元のファイル名と同じファイル名を付ける。
- (ii) 指定の区分編成ファイルの全メンバを不特定

ボリュームに退避し、指定のファイル名を付ける。

5.3.32 RESTORE文

(1) 記述形式

表 5.32

(2) 機能上の特記事項

(2.1) 出力ファイルはBACKUP文で退避する以前と同じファイルの形式で出力される。

(2.2) 既存の区分編成ファイルに追加する場合、そのブロック長、レコード長、レコード形式は元のファイルの形式と同じでなければならない。ただし、レコード形式が不定長、可変長の場合には、出力ファイルの最大ブロック長およびレコード長が元のファイルの値より大きければ一致しなくともよい。また、レコード形式が固定長の場合、出力ファイルのブロック長は、元のファイルのレコード長の倍数であれば一致していなくともよい。

(2.3) 出力ファイル名を省略した場合には、入力ファイルと同じファイル名が付けられる。

(2.4) OVを省略するとカタログ登録簿が参照される。このとき、出力ファイル名は入力ファイル名と異なるものを指定しなければならない。

(2.5) MEMパラメータを省略すると全メンバを復元する。

(3) 使用例

¥ RESTORE OF=PROG3, MEM=SUB+, OV=NS0020, NF=H01.PROG3, PSW=PS50, TRK=(10,2,RLSE)	(i)
¥ RESTORE OF=PROG2, OV=SLMT, MEM=(M1,M2,M3), NF=(H01.PROG2,AD)	(ii)

(i) バックアップ形式のファイルの指定メンバを新規作成ファイルに復元する。

(ii) バックアップ形式ファイルの指定メンバのみを、既存の区分編成ファイルに復元する。

5.3.33 DKLIST文

(1) 記述形式

表 5.33

(2) 出力情報

出力情報としてファイル名、編成、レコード形式、

レコード長、ブロック長、確保トラック数、使用中トラック数、最新参照日、他に総確保トラック数、総使用トラック数が必要である。

5.3.34 DRTYLIST文

(1) 記述形式

表 5.34

(2) 機能上の特記事項

OFパラメータで指定できるのは区分編成ファイルのみである。指定ファイルの全メンバ名を出力する。

5.3.35 MTLIST文

(1) 記述形式

表 5.35

(2) 出力情報

出力情報として指定ファイルの全ファイルについて、ファイル名、ファイル順序番号、レコード形式、レコード長、ブロック長、ブロック数が必要である。

5.3.36 F4TOX8文

(1) 記述形式

表 5.36

(2) 使用例

本マクロは次のように使用する。

```
¥ F4TO X8 H01XCHOF
```

```
    F4のジョブストリーム
```

(3), (註)

```
¥ FDDE
```

```
¥ JEND
```

5.3.37 HKJOB文, HJEND文

本マクロは次のように使用する。

```
¥ HKJOB
```

```
    F4のジョブストリーム
```

(註)

```
¥ HJEND
```

5.4 ジョブストリーム記述例

5.4.1 翻訳のみのジョブ

〔例1〕 ソースプログラムをカード入力し、翻訳を行う。この例のように、オブジェクトモジュールの保存やLIED文への引渡しがない場合は必ずNO-OBJECTオプションを指定する。この指定がないと、まったく無駄に、システムはオブジェクトモジュールを作成してしまう。

```
¥ KJOB H01XCHOF,DEBG
```

```
¥ FORTC PARAM=NOOBJECT
```

```
    ソースプログラムカード
```

```
¥ JEND
```

〔例2〕 ソースプログラムを3種類の媒体(カード, DK, MT)から入力して、翻訳を行う。この例では第一または第二のFORTC文にエラーが発見されると、ジョブ処理はそのジョブステップで終了となる。

```
¥ KJOB H01XCHOF,DEBG
```

```
¥ FORTC PARAM=NOOBJECT
```

```
    ソースプログラムカード
```

```
¥ FORTC PARAM=NOOBJECT,
```

```
    SF=(H01.A,P001)
```

```
¥ FORTC PARAM=NOOBJECT,
```

```
    SF=FORT1,SV=NM0001
```

```
¥ JEND
```

5.4.2 ユーザライブラリを作成するジョブ

〔例3〕 ソースプログラムを複数個入力して、ロードモジュール状態のユーザライブラリを作成する。

```
¥ KJOB H01XCHOF,DEBG
```

```
¥ ULIB SF=(H01.A,,H01.B,P002,
```

```
    H01.C),SINC=H01.D,
```

```
    EF=H01.L,TRK=(10,2,RLSE)
```

```
¥ JEND
```

(註) FDDE文もHJEND文も、決して省略してはならない。もし、これらの文を省略すると、その後にCRから投入された他人のカードを際限なく取込んでしまう。これに対する保全手段としては、KJOB文の前にHJEND文とFDDE文を用意しておけばよい。こうすれば、KJOB文で始まるジョブが上記のように取込まれることはない。

5.4.3 翻訳・結合編集・実行を行うジョブ

〔例4〕 ソースプログラムをカード入力して、翻訳・結合編集・実行を3ジョブステップで処理する。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC
  ソースプログラムカード
¥ LIED
¥ GO
¥ JEND

```

〔例5〕 ソースプログラムを3種類の媒体から入力して翻訳し、それら全てのオブジェクトモジュールを結合編集して実行する。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC
  ソースプログラムカード
¥ FORTCF SF=(H01.A, P300)
¥ FORTCF SF=SUB1, SV=NS0050
¥ LIED
¥ GO
¥ JEND

```

〔例6〕 ソースプログラムを個人ファイルから入力し、翻訳し、結合編集して実行する。実行時使用データはカード入力とする。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC PARAM=(ELM(*)),
      SF=(H01.MAIN, P003)
¥ LIED
¥ GO
  入力データカード
¥ JEND

```

〔例7〕 ソースプログラムを個人ファイルから複数個入力し、結合編集の出力を個人ファイルに保存するとともに実行を行う。実行時使用データは個人ファイルを参照する。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC SF=(H01.A, P300,
      H01.B, P400, H01.C, P500)

```

```

¥ LIED OEF=(H01.MAIN, OT),
      OEMEM=E1, PSW=P005,
      TRK=(10, 2)
¥ GO DF=(H01.DATA, P007),
      DMEM=(D1, D2)
¥ JEND

```

〔例8〕 実行時使用データをカード、DK, MTから論理的に一つのファイルとして入力し、実行時のFORTRAN出力を個人ファイルおよびMTに格納する。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC SF=H01.D
¥ LIED
¥ GO
  入力データカード
¥ CF DK, FILE=H01.DATA5,
      PSW=P001
¥ CF MT, FILE=DATA6, VOL=SLMT
¥ USDK U02, FILE=H01.DATA7,
      TRK=(10, 2)
¥ USMT U03, FILE=DATA8,
      VOL=SLMT
¥ JEND

```

〔例9〕 ソースプログラムをカード入力し、翻訳を行い、ローダで結合編集・実行を行う。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC
  ソースプログラムカード
¥ LINKGO ALIB=(H01.SUB9,
      P100)
¥ JEND

```

〔例10〕 ソースプログラムをカード入力し、FORTRANのGOのオプションで実行までを行う。

```

¥ KJOB H01 XCHOF, SHRT
¥ FORTC PARAM=GO
  ソースプログラムカード
¥ JEND

```

〔例11〕 ソースプログラムカードと実行時使用のカードデータを入力し、FORTRANのGOオプションで実行までを行う。

```

¥ KJOB H01XCHOF, SHRT
¥ FORTC PARAM=GO
  ソースプログラムカード
¥/
  入力データカード
¥ JEND

```

5.4.4 結合編集・実行を行うジョブ

〔例12〕 ロードモジュール状態の個人ファイルを主要入力として、2ジョブステップで結合編集し実行する。

```

¥ KJOB H01XCHOF, SHRT
¥ LIED S=ON, IEF=(H01.MAIN,
  P008), IEMEM=M1,
  ALIB=H01.SUB3
¥ GO
¥ JEND

```

〔例13〕 ロードモジュール状態の個人ファイルを主要入力として、1ジョブステップで結合編集・実行を行う。

```

¥ KJOB H01XCHOF, SHRT
¥ LINKGO EF=(H01.MAIN,
  P008), EMEM=M1,
  ALIB=H01.SUB5
¥ JEND

```

5.4.5 実行を行うジョブ

〔例14〕 実行形式のプログラムを指定して実行する。

```

¥ KJOB H01XCHOF, SHRT
¥ GO EF=(H01.A, 000P),
  EMEM=M1
¥ JEND

```

〔例15〕 複数の実行形式プログラムを実行する。

```

¥ KJOB H01XCHOF, SHRT
¥ GO EF=H01.B, EMEM=E1
  入力データカード
¥ GO EF=(H01.C, 00P0),
  EMEM=C1, DF=H01.D,
  DMEM=D1
¥ JEND

```

5.4.6 その他

〔例16〕 カードからディスクへファイルに媒体変換し、次のジョブステップでその区分編成ファイルのディレクトリに関する情報をリスト出力し、全ファイルに関する情報をリスト出力し、第一ジョブステップで作成されたファイルをFORTRANのGOオプションで実行する。

```

¥ KJOB H01XCHOF, SHRT
¥ CRTODK NF=H01.PROG3,
  PSW=P300
/ ADD MAIN, REP, LIST
  メインルーチンカード
/ ADD SUB1
  サブルーチンカード
/ ADD SUB2
  サブルーチンカード
¥ DRTYLIST OF=H01.PROG3
¥ DKLIST NAME=H01
¥ FORTC PARAM=(GO, ELM(*)),
  SF=H01.PROG3
¥ JEND

```

5.4.7 記述上の注意

FORTC文、LIED文、GO文を複数枚含むジョブストリームを記述する際、各マクロ間の関連性について次の注意が必要である。

〔例17〕 ジョブストリームの記述例

```

¥ KJOB
¥ FORTC
  ¥ FORTC
  → ¥ LIED
  → ¥ GO

```

```

┌── ¥ FORTC
└── ¥ LIED
└── ¥ GO

```

① ¥ LIED S=ON

```

┌── ¥ LIED S=ON
└── ¥ GO

```

② ¥ GO EF=H01.A, EMEM=E1
¥ JEND

〔例18〕 X8システムではFORTC文の出力、LIED文の出力は共にその保存の有無にかかわらず、それぞれLIED文、GO文に引き渡されるまで有効となっている。しかも、LIED文およびGO文には、それぞれ直前のFORTC文およびLIED文の出力が受け渡される。従って、上記の例の①、②においてパラメータの指定を省略すると、以下のよう

なマクロ間での入出力の受け渡しとなるので注意を要する。

```

¥ KJOB
┌── ¥ FORTC
└── ¥ FORTC
└── ¥ LIED
└── ¥ GO
┌── ¥ FORTC
└── ¥ LIED
└── ¥ GO
└── ¥ LIED
└── ¥ LIED S=ON
└── ¥ GO
└── ¥ GO
¥ JEND

```

表 5.1 KJOB文の記述形式

マクロ名	パラメータ	
KJOB	ユーザ登録名, ジョブクラス[,「HOLD」[,「OHOLD」] [,「TIME=実行打切時間」[,「RSTEP=処理開始ステップ番号」[,「DATE=YYMMDD」]	
パラメータ	省略値	パラメータの説明
ユーザ登録名	——	省略できない。記述形式については第5.3.1項の脚注参照。
ジョブクラス	——	ジョブクラス (DEBUG, DATA, SHRT または USER) を必ず指定する。
HOLD	——	ジョブの実行を保留したいときに指定する。但し, USERクラスを指定した場合には自動的に本パラメータがセットされる。
OHOLD	——	実行結果の出力を保留したいときに指定する。
TIME	センタ設定値	ジョブの実行打切時間を指定する。先頭に「S」をつけると秒単位,「S」の指定がないと分単位の指定となる。
RSTEP	1	処理を開始したいジョブステップ番号を指定する。1～99までの10進数で指定する。
DATE	実行日	実行中にTIMEマクロ命令を出した場合, プログラムに返される実際の実行日を変更したいとき, その日付を指定する。YY(年), MM(月), DD(日)を西暦で指定する。

表 5.2 JEND文の記述形式

マクロ名	パラメータ
JEND	パラメータなし。任意の注釈を記入できる。

表 5.3 FORTC文の記述形式

マクロ名	パラメータ	
FORTC	$[PARAM=(オプション)][,SF=(ファイル名,[パスワード]^{(注1)}...)]$ $[,SV=\left\{\begin{smallmatrix} \text{ボリューム通番, } \dots \\ \text{SLMT} \end{smallmatrix}\right\}][,SINC=(ファイル名,[パスワード]^{(注1)}...)]$ $[,RF=(ファイル名\left[\begin{smallmatrix} \text{,AD} \\ \text{,OT} \end{smallmatrix}\right])][,RMEM=(メンバ名[,R])]$ $[,RV=\left\{\begin{smallmatrix} \text{ボリューム通番, } \dots \\ \text{SLMT} \end{smallmatrix}\right\}][,PSW=(パスワード[,WRITE])][,DISP=CAT]$ $[,TRK=(トラック数[,増分][,RLSE])][,LF=(ファイル名[,OT])]$ $[,LTRK=(トラック数[,増分][,RLSE])]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	FORTTRAN 77 コンパイラオプションを指定する。
SF	——	ソースプログラムが個人ファイルまたはMTファイルに含まれる場合に、そのファイル名を指定する。多ファイルも指定可能である。
SV	——	SFで指定したファイルがMTボリューム上にある場合にそのボリューム通番を指定する。ただし、カタログされているボリュームでも省略不可であり、多ファイル多ボリュームの指定は不可能である。
SINC	——	*INCLUDE行で記述したメンバが含まれるファイルを指定する。多ファイルの指定も可能である。
RF	——	オブジェクトモジュールを個人ファイルまたはMTファイルに保存する場合、そのファイル名を指定する。
	指定なし	<p>その際のファイルの処置として、次の指定ができる。</p> <p>AD：既存のファイルに追加する指定。</p> <p>OT：新たにファイルを作成する指定。ただし、既に作成されているファイルに対してこの指定をすると、前の内容を破壊して新しいデータを書き込むので注意を要する。</p> <p>指定なし：新規ファイル作成を意味する。同名のファイルが既にあるとエラーとなる。</p>
RMEM	——	RFで指定したオブジェクトモジュールを区分編成ファイルに保存する場合にメンバ名を指定する。
RV	——	RFで指定したオブジェクトモジュールをMTボリュームに保存する場合にボリューム通番を指定する。（SVの説明参照）
PSW	——	RFで指定したファイルがパスワード保護ファイルの場合に指定する。
DISP	——	RVで指定したMTボリュームをカタログ登録する場合に指定する。
TRK	——	RFで指定したファイルが新規作成のとき、そのファイルが確保すべき領域のトラック数を指定する。ファイルの新規作成時には省略できない。
LF	——	印刷出力を個人ファイルに出力する場合に指定する。
LTRK	——	LFで指定したファイルが新規作成のとき、そのファイルが確保すべき領域のトラック数を指定する。ファイルの新規作成時には省略できない。

(注1) ファイルをパスワード保護している場合、そのパスワードを記入する。パスワード保護していない場合にはSF=(ファイル名,,ファイル名,,...)のような記述となる。

(注2) 太下線のパラメータは出来るだけ使用しないこと（第3.2.1項参照）。

表 5.4 FORTCF 文の記述形式

マクロ名	パラメータ
FORTCF	[PARAM=(オプション)][,SF=(ファイル名,[パスワード],...)] [,SV={ (ボリューム通番,...) } SLMT][,SINC=(ファイル名,[パスワード],...)] [,LF=(ファイル名[,OT])][,LTRK=(トラック数[,増分][,RLSE])]

(注) パラメータの意味は FORTC 文に同じである。

表 5.5 LIED 文の記述形式

マクロ名	パラメータ	
LIED	[PARAM=(オプション)][,S={ ON OFF}][,RF=(ファイル名,[パスワード],...)] [,RMEM=(メンバ名,...)][,RV={ (ボリューム通番,...) } SLMT] [,ALIB=(ファイル名,[パスワード],...)][,IEF=(ファイル名,[パスワード],...)] [,IEMEM=(メンバ名,...)][,OEF=(ファイル名[,OT])] [,OEMEM=(メンバ名[,R])][,PSW=(パスワード[,WRITE])] [,TRK=(トラック数[,増分][,RLSE])][,ENTRY=入口名]	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	リンケージエディタオプションを指定する。
S	OFF	先行コンパイラステップが出力したオブジェクトモジュールをリンケージエディタの主要入力とする場合には "S=OFF" を指示し、先行コンパイラステップがない場合には "S=ON" を指定する。
RF	——	既存のオブジェクトモジュールを主要入力 (S=ON) とするか、または追加入力 (S=OFF) とする場合にそのファイル名を指定する。多ファイルの指定も可能である。
RMEM	——	RF で指定したオブジェクトモジュールのメンバ名を指定する。多メンバの指定も可能である。
RV	——	RF で指定したオブジェクトモジュールが存在する MT ボリュームのボリューム通番を指定する。(FORTC 文の SV パラメータの説明参照)
ALIB	——	ロードモジュールの個人ファイルを自動呼出し入力とする場合に、そのファイル名を指定する。多ファイルの指定も可能である。
IEF	——	既存のロードモジュールの個人ファイルをリンケージエディタの入力とする場合にそのファイル名を指定する。多ファイルの指定も可能である。
IEMEM	——	IEF で指定したロードモジュールのメンバ名を指定する。多メンバの指定も可能である。
OEF	——	リンケージエディタが出力するロードモジュールを個人ファイルに保存する場合にそのファイル名を指定する。
OEMEM	——	OEF で指定したロードモジュールのメンバ名を指定する。
PSW	——	OEF で指定したファイルがパスワード保護ファイルの場合に指定する。
TRK	——	OEF で指定したファイルが新規作成の時、そのファイルが確保すべき領域のトラック数を指定する。ファイルの新規作成時には省略できない。
ENTRY	MAIN	入口プログラム名を指定する。

(注) 太下線のパラメータは出来るだけ使用しない(第 3.2.1 項参照)。

表 5.6 GO 文の記述形式

マクロ名	パラメータ	
GO	$[PARAM=(オプション)][,EF=(ファイル名[,パスワード])][,EMEM=メンバ名]$ $[,DF=(ファイル名[,パスワード])][,DMEM=(メンバ名,...)]$ $[,DV=\left\{\begin{array}{c} \text{ボリューム通番, ...} \\ \text{SLMT} \end{array}\right\}][,PAGE=出力ページ打切値]$ $[,COND=\left\{\begin{array}{c} \text{レベル番号} \\ \text{-レベル番号} \\ \text{(レベル番号1, レベル番号2)} \end{array}\right\}][,LF=(ファイル名[,OT])]$ $[,LTRK=(トラック数[,増分][,RLSE])]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	実行プログラムのオプションを指定する。
EF	——	実行プログラム（ロードモジュール）を既存の個人ファイルから入力する場合にそのファイル名を指定する。
EMEM	——	EFで指定したロードモジュールのメンバ名を指定する。
DF	——	実行時使用データが個人ファイル、またはMTファイルである場合にそのファイル名を指定する。DF,DF1,DF2,DF3,DF4の順に5個まで指定できる。
DMEM	——	DFで指定したファイルが区分編成のとき、使用データのメンバ名を指定する。DF,DF1,DF2,DF3,DF4に対して、それぞれDMEM,DMEM1,DMEM2,DMEM3,DMEM4が対応し、各々多メンバの指定が可能である。
DV	——	DFで指定したファイルがMTボリューム上にある場合にそのボリューム通番を指定する。（FORTC文のSVパラメータの説明参照）
PAGE	センタ設定値	出力結果の頁打切値を指定する。
COND	(10,49)	<p>このジョブステップを実行するかしないかの判定を以前のジョブステップ完了コードに基づいて決めるパラメータである。</p> <ul style="list-style-type: none"> COND=レベル番号 以前のジョブステップ完了コード\leqレベル番号 COND=-レベル番号 以前のジョブステップ完了コード\geqレベル番号 COND=(レベル番号1, レベル番号2) レベル番号1\leq以前のジョブステップの完了コード\leqレベル番号2 <p>条件が満たされないと、このジョブステップは迂回される。指定可能なレベル番号は10～99の10進数である。</p>

(注) 上記以外のパラメータについてはFORTC文に同じである。

表 5.7 LINKGO 文の記述形式

マクロ名	パラメータ	
LINKGO	$[PARAM=(オプション)][, \{ \frac{RF}{EF} \}=(ファイル名[, パスワード])]$ $[, \{ \frac{RMEM}{EMEM} \}=(メンバ名, \dots)][, RV=\{ \frac{(ボリューム通番, \dots)}{SLMT} \}]$ $[, ALIB=(ファイル名, [パスワード], \dots)][, DF=(ファイル名[, パスワード])]$ $[, DMEM=(メンバ名, \dots)][, DV=\{ \frac{(ボリューム通番, \dots)}{SLMT} \}]$ $[, PAGE=出力頁打切値][, COND=\left\{ \begin{array}{l} \text{レベル番号} \\ -\text{レベル番号} \\ (\text{レベル番号 1, レベル番号 2}) \end{array} \right\}]$ $[, ENTRY=入口名][, LF=(ファイル名[, OT])]$ $[, LTRK=(トラック数[, 増分][, RLSE])]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	ローダオプションを指定する。
RF EF	——	<p>{オブジェクトモジュール ロードモジュール}の個人ファイルをローダの主要入力とする時に指定する。FORTC 文の出力を入力とする場合には指定してはならない。</p> <p>{RF, RF1, RF2, RF3, RF4 EF, EF1, EF2, EF3, EF4}の順に, 5 個まで指定できる。</p>
RMEM EMEM	——	<p>{RF EF}で指定したファイルのメンバ名を指定する。{RF, RF1, RF2, EF, EF1, EF2, RF3, RF4}に対して, それぞれ {RMEM, RMEM1, RMEM2, RMEM3, EF3, EF4}が対応し, 各々多メンバの指定が可能である。</p>

(注1) 太下線のパラメータは出来るだけ使用しないこと(第 3.2.1 項参照)。

(注2) 上記以外のパラメータについては FORTC 文, LIED 文, GO 文に同じである。

表 5.8 ULIB 文の記述形式

マクロ名	パラメータ	
ULIB	EF=(ファイル名[,OT])[,FPARAM=(オプション)][,SF=(ファイル名,[パスワード],...)] [,SV={ (ボリューム通番, ...) }][,SINC=(ファイル名, [パスワード], ...)] [,LPARAM=(オプション)] [,PSW=(パスワード[,WRITE])][,TRK=(トラック数[,増分][,RLSE])] [,ENTRY=入口名]	
パラメータ	省略値	パラメータの説明
LPARAM	システム標準値	リンケージエディタオプションを指定する。
EF	——	リンケージエディタが出力するロードモジュールを個人ファイルに保存する場合に、そのファイル名を指定する。省略できない。
PSW	——	EFで指定したファイルがパスワード保護ファイルの場合、パスワードを指定する。
TRK	——	EFで指定したファイルが新規作成のとき、そのファイルが確保すべき領域のトラック数を指定する。ファイルの新規作成時には省略できない。
ENTRY	MAIN	入口プログラム名

(注) 上記以外のパラメータについては FORTC 文に同じである。

表 5.9 USMT 文の記述形式

マクロ名	パラメータ	
USMT	アクセス名[,DUMMY={ ON OFF }][,FILE=({ ファイル名 } [,FA / ,AD] [,SL [,OT] [,NL] [,ファイル順序番号])][,DISP=([CONT] [,CAT [,UNCAT]))] [,VOL={ (ボリューム通番, ...) }][,BSIZE=ブロック長] [,RSIZE=レコード長][,RECFM={ F[B] V[B] U }]	
パラメータ	省略値	パラメータの説明
アクセス名	——	ユーザのプログラムと実際の入出力装置を論理的に結合するアクセス名を必ず指定する。記述形式は Unn または FTnnF001 である。
DUMMY	OFF	実際の入出力動作が不要である場合に、DUMMY = ON を指定する。
FILE	——	SL 形式の場合、参照または作成するファイルのファイル名を指定する。 ファイル名を指定した時はファイル順序番号の省略が可能であり、ファイル名を省略した時はファイル順序番号を省略できない。 ファイル名：作成の場合は必ず指定しなければならない。 / : ファイル名を省略するときに指定する。参照の場合はファイル名の省略が可能であるが、このとき、ファイル順序番号は必須となる。 ただし、NL 形式の場合には、取り扱うファイルはファイル順序番号で指定し、ファイル名を省略する。

表 5.9 つづき

	指定なし	<p>ファイル作成の場合にファイルに対する処置を指定する。</p> <p>FA：新規ファイルを追加作成する場合に指定する。即ち、多ファイル / 単ボリュームまたは多ファイル / 多ボリュームの形式とするとときに指定する。</p> <p>AD：既存のファイルに対してデータの追加を行うとき指定する。</p> <p>OT：新たにファイルを作成する指定であり、既存のファイルに対してこの指定を行うと前の内容を破壊して作成する。</p> <p>指定なし：ボリュームの先頭から作成する。あるいはファイルの参照を意味する。</p>
	SL	<p>使用ボリュームのラベル形式を指定する。</p> <p>SL：標準ラベルボリュームを扱う指定。</p> <p>NL：ラベル無しボリュームを扱う指定。</p>
	——	<p>ファイル順序番号：磁気テープボリューム上のファイルの位置を指定する。</p>
DISP	——	<p>CONT：このマクロに割当てられているファイルをジョブ終了まで、以後のジョブステップに引き継ぐ指定。</p> <p>CAT：ファイル名とボリューム通番の対応関係をカタログ登録簿に登録する指定。</p> <p>UNCAT：ファイルをジョブから切り離すときに、指定したファイルのファイル名とボリューム通番の対応関係をカタログ登録簿から削除する指定。</p>
VOL	——	<p>SL形式の場合に、ボリューム通番または不特定ボリュームを指定する。既にカタログ登録してあるファイルを使用する場合は省略可能である。</p> <p>ボリューム通番：6文字の英数字で予めセンタで一意的ラベル名を書き込む。</p> <p>SLMT：標準ラベルをもつ磁気テープボリュームを使用する指定（不特定ボリュームの指定）。</p>
BSIZE	SL形式 (FORTRAN	<p>ブロック長を指定する。単位はバイト数である。SL形式のボリュームを参照するときには指定してはならない。</p>
RSIZE	コンパイラの 標準値)	<p>レコード長を指定する。単位はバイト数である。SL形式のボリュームを参照するときには指定してはならない。</p>
RECFM	NL形式 (省略不可)	<p>レコード形式を指示する。SL形式のボリュームを参照するときには指定してはならない。</p> <p>F：固定長の指定。</p> <p>V：可変長の指定。</p> <p>U：不定長の指定。</p> <p>B：ブロック化レコードの指定。</p>

表 5.10 USDK 文の記述形式

マクロ名	パラメータ	
USDK	アクセス名 [, DUMMY = { $\begin{smallmatrix} \text{ON} \\ \text{OFF} \end{smallmatrix}$ }] [, FILE = (ファイル名 [, $\begin{smallmatrix} \text{AD} \\ \text{OT} \end{smallmatrix}$])] [, DISP = ([CONT] [, DLT])] [, MEM = (メンバ名 [, R])] [, PSW = (パスワード [, WRITE])] [, { $\begin{smallmatrix} \text{TRK} = (\{ \text{トラック数} \} \\ \text{BLK} = (\text{ブロック長} * \text{ブロック数}) \end{smallmatrix}$ } [, 増分] [, RLSE])] [, BSIZE = ブロック長] [, RSIZE = レコード長] [, RECFM = { $\begin{smallmatrix} \text{F} [\text{B}] \\ \text{V} [\text{B}] \\ \text{U} \end{smallmatrix}$ }]]	
パラメータ	省略値	パラメータの説明
アクセス名	——	ユーザのプログラムと実際の入出力装置を論理的に結合するアクセス名を必ず指定する。記述形式は USMT 文に同じ。
DUMMY	OFF	実際の入出力動作が不要である場合に、DUMMY = ON を指定する。
FILE	——	ファイル名：参照または作成するファイル名を指定する。
	指定なし	ファイル作成の場合にファイルに対する処置を指定する。 AD：既存の順編成ファイルに対してはデータの追加を行う指定。既存の区分編成ファイルに対しては既存メンバの最終位置から新メンバを作成する指定。 OT：ファイルを新規作成する指定。指定ファイルが既に存在すると、前の内容を破壊して新しいデータを書き込む。 指定なし：ファイルの参照またはファイルの新規作成を意味する。後者の場合は領域確保パラメータの指定が必須である。作成の場合、指定ファイルが既に存在するとエラーとなる。
DISP	——	CONT：このマクロに割当てられているファイルをジョブ終了まで、以後のジョブステップに引き継ぐ指定。
	KEEP	DLT：ジョブステップ終了時に指定した保存ファイルの消去を行う指定。同時にカタログ登録簿からも削除される。
MEM	——	追加作成または置き換えをするメンバ名を指定する。 メンバ名：先頭が英字で始まる 8 文字以内の英数字。 R：メンバを置き換える場合に指定。

表 5.10 つづき

PSW	_____	<p>ファイルをパスワード保護する場合、パスワード保護ファイルを参照する場合に指定する。</p> <p>パスワード：4文字の英数字。</p> <p>WRITE　：WRITE保護ファイルとする指定。</p> <p>詳細は次のとおり</p> <table><tr><th>ファイル作成時の PSWパラメータ</th><th>機　　　能</th></tr><tr><td>指定なし</td><td>参照／更新／削除とも保護されない</td></tr><tr><td>PSW=パスワード</td><td>参照／更新／削除に対して保護が行われる。参照／更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態を READ /WRITE保護という。</td></tr><tr><td>PSW=(パスワード, WRITE)</td><td>更新／削除に対して保護が行われる。参照の場合にはパスワードを指定する必要はない。更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態をWRITE保護という。</td></tr></table>	ファイル作成時の PSWパラメータ	機　　　能	指定なし	参照／更新／削除とも保護されない	PSW=パスワード	参照／更新／削除に対して保護が行われる。参照／更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態を READ /WRITE保護という。	PSW=(パスワード, WRITE)	更新／削除に対して保護が行われる。参照の場合にはパスワードを指定する必要はない。更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態をWRITE保護という。
ファイル作成時の PSWパラメータ	機　　　能									
指定なし	参照／更新／削除とも保護されない									
PSW=パスワード	参照／更新／削除に対して保護が行われる。参照／更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態を READ /WRITE保護という。									
PSW=(パスワード, WRITE)	更新／削除に対して保護が行われる。参照の場合にはパスワードを指定する必要はない。更新／削除を行う場合にはパスワードを正しく指定しなければならない。このときの保護状態をWRITE保護という。									
TRK または BLK	_____	<p>トラック数：確保すべきトラック数を指定。</p> <p>ブロック長：1ブロックの大きさをバイト単位で指定。</p> <p>ブロック数：確保すべきブロック数を指定。</p> <p> / ：既存のファイルに対して増分または RLSEを指定するときの記号。</p> <p>増分　　：領域が不足した場合の追加量を指定する。TRKを指定した場合、増分はトラック数、BLKを指定の場合にはブロック数で指定。</p> <p>RLSE　：使用されなかった領域の解放を行う指定。</p>								
BSIZE	FORTRAN コンパイラの 標準値	ファイル作成時、ブロック長を指定する。								
RSIZE		ファイル作成時、レコード長を指定する。								
RECFM		ファイル作成時、レコード形式を指定する。								
		F：固定長の指定。 V：可変長の指定。 U：不定長の指定。 B：ブロック化レコードの指定。								

表 5.11 USWK 文の記述形式

マクロ名	パ ラ メ ー タ
USWK	<p>アクセス名 [, DUMMY = { ON OFF }], { TRK = (トラック数 BLK = (ブロック長 * ブロック数) [, 増分] [, RLSE])</p> <p>[, BSIZE = ブロック長] [, RSIZE = レコード長] [, RECFM = { F[B] V[B] U }]</p>

(注) パラメータの意味は USDK 文に同じである。

表 5.12 CF 文の記述形式

マクロ名	パラメータ	
CF	装置名 [, FILE = ({ ファイル名 } [, SL] [, NL] [, ファイル順序番号])] [, VOL = { (ボリューム通番 , ...) }] [, MEM = (メンバ名 , ...)] [, PSW = パスワード] [, BSIZE = ブロック長] [, RSIZE = レコード長] [, RECFM = { $\begin{matrix} F[B] \\ V[B] \\ U \end{matrix}$ }]	
パラメータ	省略値	パラメータの説明
装置名	——	連結するファイルを入力する装置名で必ず指定する。指定可能な装置は DK , MT , CRである。

(注) その他のパラメータの意味はUSMT文とUSDK文に同じである。

表 5.13 SAME 文の記述形式

マクロ名	パ ラ メ ー タ
SAME	<p>アクセス名, FILE = ({ ファイル名 } [, FA] [, SL] [, ファイル順序番号]) [, AD] [, NL] [, *] [, OT]</p> <p>[, DISP = ([CAT] [, REF]) [, UNCAT] [, VOL = { (ボリューム通番, ...) }] SLMT</p> <p>[, BSIZE = ブロック長] [, RSIZE = レコード長] [, RECFM = { F [B] } V [B] } U]</p>

(注) “*”は現時点の磁気テープボリューム上の位置より処理を行う指定。“REF”はカタログ登録簿からボリューム通番を参照する指定。その他のパラメータの意味はUSMT文に同じである。

表 5.14 XY の記述形式

マクロ名	パ ラ メ ー タ
XY	[DEBUG]

(注) DEBUGパラメータはデバッグのみで実際にXYに出力しない場合に指定する。

表 5.15 UPDATE 文の記述形式

マクロ名	パラメータ	
UPDATE	OF=(ファイル名[,パスワード])[,PARAM=(オプション)] [,MEM=メンバ名][,OV={ (ボリューム通番, …) }][,NV={ (ボリューム通番, …) }]	
パラメータ	省略値	パラメータの説明
PARAM	(2.8) 参照	LIBEのUPDATEのオプションを指定する。
OF	——	更新するファイル名を必ず指定する。
MEM	——	ファイルが区分編成のとき, 更新するメンバ名を指定する。
OV	——	更新するファイルが磁気テープ上にある場合, そのボリューム通番を指定する。NL, NSL形式は使用不可能である。なお, カタログされているボリュームでも省略不可である。
NV	——	更新したファイルを磁気テープ上に格納する場合にそのボリューム通番を指定する。なお, ボリューム通番またはSLMTを必ず指定しなければならない。

表 5.16 INSERT 文の記述形式

マクロ名	パラメータ	
INSERT	IDSEQ= { 識別順序 } TOP BOT [,MEM=メンバ名][,INIT=初期値][,INCR=増分値]	
パラメータ	省略値	パラメータの説明
IDSEQ	——	レコードを挿入する位置を必ず指定する。 識別順序: 指定された識別順序をもつレコードの後に追加データを挿入する指定。 TOP : 更新するファイルの先頭レコードの前に追加データを挿入する指定。 BOT : 更新するファイルの最後のレコードの後に追加データを挿入する指定。
INIT	——	挿入するレコードに識別順序を付加するときの初期値を指定する。
INCR	1	挿入するレコードに識別順序を付加するときの増分値を指定する。最大は3ケタの10進数である。

(注) 上記以外のパラメータの意味はUPDATE文に同じである。

表 5.17 MERGE 文の記述形式

マクロ名	パラメータ
MERGE	OF=(ファイル名[,パスワード])[,MEM=メンバ名][,PARAM=(オプション)]

(注) パラメータの意味はUPDATE文に同じである。

表 5.18 OMIT文の記述形式

マクロ名	パラメータ	
OMIT	IDSEQ={ 識別順序 1 } [: { 識別順序 2 }], OF=(ファイル名 [, パスワード]) [, MEM=メンバ名] [, PARAM=(オプション)]	
パラメータ	省略値	パラメータの説明
IDSEQ	——	削除するレコードを必ず指定する。 TOP：先頭レコードから削除する指定。 BOT：最終レコードまで削除する指定。 識別順序 1：この識別順序から削除する指定。 識別順序 2：この識別順序まで削除する指定。

(注) 上記以外のパラメータの意味はUPDATE文に同じである。

表 5.19 RENUM文の記述形式

マクロ名	パラメータ
RENUM	OF=(ファイル名 [, パスワード]) [, IDSEQ={ 識別順序 1 } [: { 識別順序 2 }]] [, PARAM=(オプション)] [, MEM=メンバ名] [, INIT=初期値] [, INCR=増分値]

(注) パラメータの意味はINSERT文、OMIT文に同じである。ただし、IDSEQパラメータの省略値はTOP:BOTであり、INCRパラメータの省略値は10である。

表 5.20 DELETE文の記述形式

マクロ名	パラメータ	
DELETE	OF=(ファイル名 [, パスワード]) [, PARAM=(オプション)] [, MEM={ (メンバ名, ...) }]	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBEのSCRACHのオプションを指定する。ただし、MEMパラメータ指定の場合のみ有効である。
OF	——	消去するファイルのファイル名を必ず指定する。
MEM	——	区分編成ファイルの消去するメンバ名を指定する。 (メンバ名, ...)：消去するメンバ名を指定。 メンバ名[:メンバ名]：連続して消去する先頭と最終のメンバ名を指定。 文字列+：指定文字列から始まる全メンバの消去を指定。 +：全メンバの消去を指定。

表 5.21 CRTODK 文の記述形式

マクロ名	パラメータ	
CRTODK	$NF = (\text{ファイル名} \begin{bmatrix} ,AD \\ ,OT \end{bmatrix}) [,ORG = \{ \begin{matrix} PO \\ PS \end{matrix} \}] [,PARAM = (\text{オプション})]$ $[,PSW = (\text{パスワード} [,WRITE])] [,TRK = (\text{トラック数} [,増分] [,RLSE])]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の ADD のオプションを指定する。
NF	——	個人ファイルのファイル名を必ず指定する。
ORG	PO	出力ファイルの編成を指定する。 PO : 区分編成ファイルとする指定。 PS : 順編成ファイルとする指定。
PSW	——	NF で指定したファイルがパスワード保護ファイルのとき指定する。
TRK	——	確保すべき領域の大きさを指定する。ファイルの新規作成時には必須パラメータである。

表 5.22 CRTOMT 文の記述形式

マクロ名	パラメータ	
CRTOMT	$[PARAM = (\text{オプション})] [,NF = (\text{ファイル名} \begin{bmatrix} ,FA \\ ,AD \\ ,OT \end{bmatrix})]$ $[,NV = \{ (\text{ボリューム通番}, \dots) \}]$ SLMT	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の ADD のオプションを指定する。
NF	——	磁気テープに出力するファイル名を指定する。このパラメータを省略して登録カードの先頭に補助制御文を挿入してもよい。
NV	SLMT	出力する磁気テープのボリューム通番を指定する。

表 5.23 CRTOLP 文の記述形式

マクロ名	パラメータ	
CRTOLP	$[PARAM = (\text{オプション})] [,COPY = \text{複写部数}]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の PRINT のオプションを指定する。
COPY	1	印刷出力の複写部数を指定する。

表 5.24 DKTODK 文の記述形式

マクロ名	パラメータ	
DKTODK	OF = (ファイル名[, パスワード])[, PARAM = (オプション)] , NF = (ファイル名[, AD][, OT])[, MEM = { (メンバ名, ...) メンバ名[:メンバ名] 文字列 + + } [, ORG = { PS PO }][, PSW = (パスワード[, WRITE])] [, TRK = (トラック数[, 増分][, RLSE])	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の COPY のオプションを指定する。
OF	——	入力側の個人ファイル名を必ず指定する。
NF	——	出力側の個人ファイルのファイル名を必ず指定する。
ORG	——	出力ファイル編成を入力ファイルと異編成にする指定 PS : 区分編成を順編成にする指定。 PO : 順編成を区分編成にする指定。

(注) 上記以外のパラメータの意味は DELETE 文と CRTODK 文に同じである。

表 5.25 DKTOMT 文の記述形式

マクロ名	パラメータ	
DKTOMT	OF = (ファイル名[, パスワード])[, PARAM = (オプション)] [, MEM = { (メンバ名, ...) メンバ名[:メンバ名] 文字列 + + } [, NF = (ファイル名[, FA][, OT])] [, NV = { (ボリューム通番, ...) SLMT } 	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の COPY のオプションを指定する。
OF	——	個人ファイルのファイル名を必ず指定する。順編成ファイルの場合には OF, OF1, OF2, OF3, OF4 の順に多ファイルの指定ができる。
MEM	——	指定 OF が区分編成ファイルの場合、メンバ名を指定する。
NF	——	MT ファイルのファイル名を指定する。入力する各メンバを順編成ファイルとする場合には指定してはならない。順編成の多ファイルを入力した場合、OF, OF1, OF2, OF3, OF4 に対応して NF, NF1, NF2, NF3, NF4 を指定しなければならない。
NV	SLMT	出力磁気テープのボリューム通番を指定する。

(注) 上記以外のパラメータの意味は DELETE 文と CRTOMT 文に同じである。

表 5.26 DKTOLP 文の記述形式

マクロ名	パラメータ	
DKTOLP	OF=(ファイル名[,パスワード])[,PARAM=オプション][,COPY=複写部数] [,MEM= $\left\{ \begin{array}{l} (\text{メンバ名}, \dots) \\ \text{メンバ名}[:\text{メンバ名}] \\ \text{文字列} + \\ + \end{array} \right\}$][,RECNO= $\left\{ \begin{array}{l} \text{TOP} \\ \text{相対番号 1} \end{array} \right\}[:\left\{ \begin{array}{l} \text{BOT} \\ \text{相対番号 2} \end{array} \right\}]]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の PRINT のオプションを指定する。
COPY	1	印刷出力の複写部数を指定する。
OF	——	印刷出力を行う個人ファイルのファイル名を必ず指定する。
RECNO	TOP: BOT	指定のファイルまたはメンバ内の特定レコードの印刷出力を指定する。 TOP: 先頭レコードからの指定。 BOT: 最終レコードまでの指定。 相対番号 1: この番号からの指定。 相対番号 2: この番号までの指定。 なお、相対番号 1 < 相対番号 2 で、番号は最大 5 ケタの 10 進数である。

(注) 上記以外のパラメータの意味は DELETE 文に同じである。

表 5.27 MTTODK 文の記述形式

マクロ名	パラメータ	
MTTODK	OF=($\left\{ \begin{array}{l} \text{ファイル名} \\ / \end{array} \right\} [, \text{SL}] [, \text{NL}] [, \text{ファイル順序番号}]) [, \text{PARAM} = (\text{オプション})]$ [,OV= $\left\{ \begin{array}{l} (\text{ボリューム通番}, \dots) \\ \text{SLMT} \end{array} \right\}$],NF=($\left\{ \begin{array}{l} \text{ファイル名} \\ [, \text{AD}] \\ [, \text{OT}] \end{array} \right\}$) [,ORG= $\left\{ \begin{array}{l} \text{PO} \\ \text{PS} \end{array} \right\}$][,PSW=(パスワード[,WRITE])] [,TRK=(トラック数[,増分][,RLSE])][,BSIZE=ブロック長] [,RSIZE=レコード長][,RECFM= $\left\{ \begin{array}{l} \text{F[B]} \\ \text{V[B]} \\ \text{U} \end{array} \right\}$]	
パラメータ	省略値	パラメータの説明
OF	——	入力する MT ファイルのファイル名を必ず指定する。順編成のときは OF, OF1, OF2, OF3, OF4 の順に多ファイルの指定ができる。このときにはそれぞれ NF, NF1, NF2, NF3, NF4 を対応させる。
OV	——	入力する MT ボリュームのボリューム通番を指定する。
BSIZE	——	NL 形式の磁気テープを入力するとき、ブロック長を指定する。
RSIZE	——	NL 形式の磁気テープを入力するとき、レコード長を指定する。
RECFM	——	NL 形式の磁気テープを入力するとき、レコード形式を指定する。

(注) 上記以外のパラメータの意味は DKTODK 文に同じである。

表 5.28 MTTOMT文の記述形式

マクロ名	パラメータ	
MTTOMT	$[PARAM=(\text{オプション})][,OFS=(\{ \text{ファイル名} \} [,SL] [,NL] [, \text{ファイル順序番号}])]$ $[,OFM=\left\{ \begin{array}{l} (\text{ファイル名}, \dots) \\ (\{ \text{ファイル名} \}_* [: \{ \text{ファイル名} \}_*]) \\ + \\ n \end{array} \right\}]$ $[,NFS=(\{ \text{ファイル名} \} [,FA] [,AD] [,OT] [, \text{ファイル順序番号}])]$ $[,OV=\{ \text{ボリューム通番}, \dots \}_{SLMT}] [,NV=\{ \text{ボリューム通番}, \dots \}_{SLMT}]$ $[,BSIZE=\text{ブロック長}] [,RSIZE=\text{レコード長}] [,RECFM=\left\{ \begin{array}{l} F[B] \\ V[B] \\ U \end{array} \right\}]$	
パラメータ	省略値	パラメータの説明
OFS	——	MT ファイルを単ファイル入力する場合に指定する。
OFM	——	<p>MT ファイルを多ファイル入力する場合に指定する。この場合、出力ファイル名を指定することができず、自動的に入力ファイル名と同じになる。</p> <p>(ファイル名, …) : 複写するファイル名を指定する。</p> <p>($\{ \text{ファイル名} \}_*$ [: $\{ \text{ファイル名} \}_*$]) : 指定のファイルからファイルまでの複写する区間を指示する。*は先頭または最終ファイルを意味する。</p> <p>+ : 全ファイルの複写を指示する。</p> <p>n : 先頭から n 番目までのファイルの複写を指示する。</p>
NFS	——	入力が単ファイルの場合に、出力ファイル名を指定する。
NV	SLMT	出力MTボリュームのボリューム通番を指定する。

(注) 上記以外のパラメータの意味はMTTODK文に同じである。

表 5.29 MTTOLP文の記述形式

マクロ名	パラメータ	
MTTOLP	$OF=(\{ \text{ファイル名} \} [,SL] [,NL] [, \text{ファイル順序番号}])$ $[,PARAM=(\text{オプション})][,COPY=\text{複写部数}] [,OV=\{ \text{ボリューム通番}, \dots \}_{SLMT}]$ $[,RECNO=\{ \text{TOP 相対番号 1} \} [: \{ \text{BOT 相対番号 2} \}]] [,BSIZE=\text{ブロック長}]$ $[,RSIZE=\text{レコード長}] [,RECFM=\left\{ \begin{array}{l} F[B] \\ V[B] \\ U \end{array} \right\}]$	

(注) パラメータの意味はDKTOLP文とMTTODK文に同じである。

表 5.30 MTCOPY 文の記述形式

マクロ名	パラメータ	
MTCOPY	$\left[\text{PARAM} = \left\{ \begin{matrix} (*) \\ \text{H[EAD]} \\ \text{A[FTER]} \end{matrix} \right\} \right] \left[, \text{DEN} = \left\{ \begin{matrix} 800 \\ 1600 \end{matrix} \right\} \right] \left[, \text{OV} = \left\{ \begin{matrix} \text{ボリューム通番} \\ \text{SLMT} \end{matrix} \right\} \right]$ $\left[, \text{OV1} = \text{ボリューム通番} \right] \left[, \text{OV2} = \text{ボリューム通番} \right] \left[, \text{OV3} = \text{ボリューム通番} \right]$ $\left[, \text{OF} = (/ , \left\{ \begin{matrix} \text{SL} \\ \text{NL} \end{matrix} \right\}) \right] \left[, \text{NV} = \text{ボリューム通番} \right]$	
パラメータ	省略値	パラメータの説明
PARAM	(*)	<p>デッドコピーおよびコピーの条件を指定する。</p> <p>(*)：入力磁気テープのファイルを全て出力磁気テープヘッドコピーする指定。</p> <p>H[EAD]：複数個の入力磁気テープの全ファイルを単一の出力磁気テープの先頭から書き込む指定。</p> <p>A[FTER]：複数個の入力磁気テープの全ファイルを単一の出力磁気テープの既書き込み済みのファイルの後からコピーする指定。</p>
DEN	1600	出力磁気テープの記録密度を指定する。記録密度は 800BPI, または 1600BPI である。なお, AFTER を指定した場合, DEN 指定は意味を持たず, 既書かれている記録密度となる。
OV[n]	——	入力磁気テープのボリューム通番を指定する。デッドコピーの場合は省略できる。
OF	(/ , SL)	入力, 出力磁気テープのラベル形式を指定する。
NV	——	出力磁気テープのボリューム通番を指定する。省略した場合は OV と同じになる。

表 5.31 BACKUP 文の記述形式

マクロ名	パラメータ	
BACKUP	$\text{OF} = (\text{ファイル名} [, \text{パスワード}]) [, \text{PARAM} = (\text{オプション})]$ $\left[, \text{MEM} = \left\{ \begin{matrix} (\text{メンバ名} , \dots) \\ \text{メンバ名} [: \text{メンバ名}] \\ \text{文字列} + \\ + \end{matrix} \right\} \right]$ $\left[, \text{NF} = (\text{ファイル名} [, \text{FA}] [, \text{OT}]) \right] \left[, \text{NV} = \left\{ \begin{matrix} (\text{ボリューム通番} , \dots) \\ \text{SLMT} \end{matrix} \right\} \right]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBE の BACKUP のオプションを指定する。
OF	——	退避する区分編成ファイルのファイル名を指定する。
NF	——	磁気テープボリューム上のファイル名を指定する。
NV	SLMT	磁気テープのボリューム通番を指定する。

(注) 上記以外のパラメータの意味は DELETE 文に同じである。

表 5.32 RESTORE文の記述形式

マクロ名	パラメータ	
RESTORE	$OF = \text{ファイル名} [, PARAM = (\text{オプション})] [, MEM = \left\{ \begin{array}{l} (\text{メンバ名}, \dots) \\ \text{メンバ名} [: \text{メンバ名}] \\ \text{文字列} + \\ + \end{array} \right\}]$ $[, OV = \left\{ \begin{array}{l} (\text{ボリューム通番}, \dots) \\ \text{SLMT} \end{array} \right\}] [, NF = (\text{ファイル名} [, AD] [, OT])]$ $[, PSW = (\text{パスワード} [, WRITE])] [, TRK = (\text{トラック数} [, 増分] [, RLSE])]$	
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	LIBEのRESTOREのオプションを指定する。
OF	——	復元する磁気テープボリューム上のバックアップ形式のファイル名を必ず指定する。

(注) 上記以外のパラメータの意味はDELETE文、MTTODK文に同じである。

表 5.33 DKLIST文の記述形式

マクロ名	パラメータ
DKLIST	NAME = ユーザ名

(注) ユーザ名は必ず指定する。ユーザ名の記述形式は第 5.3.1 項の脚注の通り。

表 5.34 DRTYLIST文の記述形式

マクロ名	パラメータ
DRTYLIST	OF = (ファイル名 [, パスワード])

(注) 区分編成のファイル名を必ず指定する。

表 5.35 MTLIST文の記述形式

マクロ名	パラメータ
MTLIST	$VOL = \left\{ \begin{array}{l} \text{ボリューム通番} \\ \text{SLMT} \end{array} \right\}$

(注) VOLパラメータの省略値はSLMTである。

表 5.36 F4TOX8文の記述形式

マクロ名	パラメータ
F4TOX8	ユーザ登録名

6. F4 システム用カタプロの設計

OSIV/F4 システムは大型の計算機システムに適した OS といえよう。大型計算機の場合は小型の場合と違って、ユーザは直接計算機を操作できず、オープン機器によりバッチジョブの投入と取出しを行い、TSS 端末より会話型処理を行う。多数のユーザが容易に大型計算機システムを利用でき、かつ融通性に富んだシステム運用を実現する上で、次の 3 点が重要である。

第一に、バッチジョブの投入手続きと処理結果の取出し手続きが容易なことである。即ち、カタプロ（マクロと呼ばない理由は脚注）記述形式が簡明であり、ジョブストリームが容易に記述でき、かつ機器の操作が簡単なことが要求される。

第二に、会話型処理機能が十二分に活用できることである。即ち、本来 TSS 端末から行うべき処理は応答性の非常に悪いバッチ処理に依存する必要のなきことが要求される。

第三は磁気テープを使用するジョブ処理におけるユーザの不便を解消することである。即ち、現用システムでは磁気テープ装置を使用する場合、ユーザはそのジョブの実行に立ち合い、ジョブ処理の開始・終了時に磁気テープの取付け・取外しを行わなければならない。しかも、ジョブ処理の開始・終了時間はともに不明であるから、ユーザの不便は甚だしい。この第三の問題は、もし十分なディスク容量をシステムに装備できれば、著者らが文献(9)で提案した方策により完全に克服できる。

次期大型計算機システムでは是非上記 3 つの問題を解決すべきであり、また文献(9)で提示したシステム構成とシステム運用法が実現できれば、十分に解決することができる。そこで、F4 システム用カタプロは文献(9)で示したシステム運用上、必要かつ十分なもののみの設計する。

始めに、第 4 章で示したカタプロを、システム運用面から要・不要の整理を行うとともに、いくつかの新規機能について説明しておく。

6.1 カタプロ設計について

第 4 章で示したカタプロの内、運用上ユーザコンソールコマンドとすべきもの⁽⁹⁾、申請ジョブのみ使用可能なもの⁽⁹⁾および不要なものは次の通りである。これら以外のカタプロがオープンバッチジョブの投入手続きとして使用できる。

(1) ユーザコンソールコマンドとすべきカタプロ
次のものはユーザコンソールコマンドとする。

MTTODK 文, DKTOMT 文, FPTODK 文,
DKTOFP 文, BACKUP 文, RESTORE 文,
MTLIST 文

もしこれらのカタプロを用意し、これらをバッチ処理の対象とすると、前記の第三の問題は全く解決されない。そこで、ユーザコンソールという新規機能の装置を設置し、それらを TSS の配下で動作させるならば、メニュー画面上のコマンド名をライトペンで選択し、システムが出す指示に従うことにより、ユーザは迅速に必要とする処理を終了することができるから、格段とユーザの便宜が計られる。

(2) 申請ジョブ用カタプロ

次のものは申請ジョブのみが使用可能なものとする。

USMT 文, (MT を使用する) CF 文, MTTOMT 文, MTCOPY 文

前 2 者は実行時に磁気テープへのアクセスを要求する文であり、後 2 者は磁気テープ装置を 2 台要求する文であるから、システム資源管理の使用制限上、申請ジョブでしか使用できない。

(3) 不要カタプロ

次のものは F4 システムでは不要である。

CRTOMT 文, MTTOLP 文, DELETE 文,
F4TOX8 文, HKJOB 文, HJEND 文

前 2 者のように、オープン機器複数台の要求はシステム資源管理の使用制限上禁止する。第三番目のものは TSS 端末またはユーザコンソールから行う。また、UPDATE 文も出来るだけ不要とし、データセットの編集作業は TSS 端末から処理すべきである。

(注) F4 システムではマクロ定義体をカタログドブロンジャ（略してカタプロ）と呼ぶので、以後マクロ名とはいわずにカタプロ名と呼ぶ。

(4) 追加カタプロ

F4 システムの利用上の便宜のため、次のカタプロを追加する。

COMPR文, MTRW文

なお、詳細については後述する。

(5) MTスプール機能

前記第三の問題の内、媒体変換処理に関しては(1)により完全に解決される。残る問題、ユーザプログラム実行時の磁気テープへのアクセスに関する問題はMTスプール機能により解決する。即ち、ユーザがMTR文を用意すれば、実行時参照の磁気テープ上のデータはジョブ投入手続きの一環として、システムがディスク上に格納する。ユーザはジョブ投入時にシステムが出す指示に従って、磁気テープの取付け・取外しを行えばよく、ジョブの実行に立合う必要がない。また、ユーザがMTW文を用意すれば、システムは実行時に作成される磁気テープデータを一たんディスク上に格納しておき、ユーザが磁気テープ上に取出すのを待つ。ユーザはジョブの実行に立合う必要がなく、ジョブの終了を確認してから取出せばよい。

以上の処理は(1)と同様迅速に終了するから、前記第三の問題は完全に解消される。なお当然ながら、オープンの磁気テープ装置に対しては実行時アクセスが禁止される。やむをえず磁気テープへの実行時アクセスを必要とする場合には、そのジョブを申請ジョブとすれば、USMT文が使用でき、磁気テープへの実行時アクセスが可能である。

(6) 個人ファイルの機密保護

システムは全ての個人ファイルを厳重に機密保護する。機密保護の実現手段にはパスワード照合による方法と、アクセス権設定による方法とがあるが、次期システムでは後者の手段を採用すべきである。その場合、カタプロ上にデータセットの機密保護に関するパラメータが一切必要なく、カタプロ記述形式も簡明化される。なお、個人ファイルに対するアクセス権は各自の責任で、TSS端末から他人に開放していくことができる。

(7) ユーザジョブ処理自動継続機能

システム運用上、ジョブクラス毎に中央処理装置使用制限時間を設けなければならない。しかし、こ

の制限値が最大のジョブクラスで実行しても、制限時間内で処理を終了しえないジョブもある。このようなジョブ処理の継続をユーザプログラムで対処させることなく、カタプロのパラメータで指示すれば、システムが自動的にジョブ処理の実行継続に必要な処置をとる機能を用意し、ユーザへの便宜を計る。

(8) カタプロ記述形式の簡明化

上記(1), (2), (5)の機能により、翻訳・結合編集・実行などバッチジョブの中核をなすカタプロ上に、磁気テープ使用を指示するパラメータは不要になる。こうして、基本的なカタプロで行うデータセットの作成・参照は個人ファイルに限定できる。さらに、個人ファイルに作成されるデータセットを全て自動的にカタログ登録する機能を用意すれば、装置のボリューム番号とグループ名を指示するパラメータをも不要なものとしえる。また、TSS端末から実行結果の検索ができる機能を用意すれば、ジョブステップ毎に出力先を指定するパラメータは不要となり、ジョブ毎の出力先指定ができれば十分である。

次期システムではこれら全ての機能を用意し、それによりカタプロ記述形式も出来る限りの簡明化を計る。

6.2 ジョブ制御マクロ機能

F4システムにも、MONITORⅦシステム以上でX8システム相当のジョブ制御機能はある。しかし、後2者と比較して、次のような使用法上の相違点といくつかの制限事項とがある。

(1) カタプロ呼出し方法の相違

MONITORⅦ, X8システムではジョブ制御文のオペレーション欄でマクロ名を指定すれば、必要とするマクロを呼出すことができる。しかし、F4システムでのカタプロの呼出しにはジョブ制御文のオペレーション欄をEXECまたはEXPANDとしなければならない。また、カタプロ名はそのオペランド欄の先頭パラメータとして指定しなければならない。また、EXEC文で呼出せるカタプロ名はEXEC文で始まるカタプロに限られる。EXEC文で始まっていないか、全くEXEC文を含まないカタプロを呼出す場合にはEXPAND文を使用しなければならない。

(2) 制限事項

MONITORⅦ, X8 システムと異なり, F4 システムではカタプロの中にジョブ識別用制御文を含めることができないため, 第 4.1 項で示した KJOB 文と JEND 文が作成できず, JOB 文と空文というシステムのジョブ制御文そのものを使用しなければならない。

また, F4 システムではジョブストリームの入力処理時に, ジョブ制御文とそれ以外のカード入力データとを区分けしてディスク上に格納する方式がとられている。このため, ジョブストリームの中でジョブ制御文以外のカード入力データがあると, その直前にその旨を指示するジョブ制御文(DD文)の挿入が必要である。しかし, カタプロの中にジョブ制御文以外のカード入力を指示する DD 文と, それに続くデータとを含めることができない。よって, ジョブ制御文以外のカード入力を指示する DD 文はユーザがジョブストリームの中で用意しなければならないが, 省略の可否および記述形式については第 6.4 項で詳述する。

6.3 全体的注意事項

6.3.1 データセット名について

(1) 個人ファイル上のデータセット名

個人ファイル上のデータセット名は 44 文字以内で, かつ, 次の条件を満たしていなければならない。

- データセット名は英字で始まる 8 文字以内の英数字の文字列(修飾子と呼ぶ)をピリオドで結んだものであること。ただし, 第 1 修飾子は規約通りのユーザ名, グループ名等であること。

(2) MT 上のデータセット名

MT 上のデータセット名は 17 文字以内で, かつ第 1 修飾子の規約をのぞく上記の条件を満たしていなければならない。17 文字以上の名前を指定すると後の 17 文字が有効となる。

6.3.2 記述上の注意

(1) 一般的事項

各カタプロのパラメータ記述形式は第 6.4 項で示す書き方に従わなければならない。なお, 下記の事項を除き, 記述上の全体的取り決めは第 5.2.2 項の(1)に同じ。

- 全てのカタプロ文は次のカードに継続できるが,

その際必ず“, ”記号で記述が終っていなければならない。また, 継続行の第一, 二桁には記号“//”を入れ, 三桁目は空白とし, 四桁目以降から継続するオペランドを記述しなければならない。

- 個人ファイルのデータセットは“データセット名[(メンバ名)]”の形式で指定できる。ここで, メンバ名を指定すると区分編成とみなされ, 省略すると順編成とみなされる。

(2) 特殊事項

第 6.4 項で示す記法の内, 一部のみ次の書き方も可能である。

- キーワードパラメータの右边を引用符で囲んで示しているものの内, 右边の中に特殊記号を含まない記述となる場合には引用符を省略できる。

6.3.3 個人ファイルの作成・参照について

下記の事項を除き, 第 5.2.3 項に同じ。

(1) 定義可能なデータセット識別番号 00～99 の内, 05 はカードの読み込み, 06 はリスト出力, 07 はカード出力, 98 はグラフィックディスプレイ出力, 99 はプリンタプロッタおよび XY 出力のためにシステムが使用する。なお, 01 もシステムが使用する場合がある。

(2) F4 システムではデータセットの機密保護のためのパスワードは不要である(第 6.1 項の(6)参照)。

6.3.4 データセットの処置について

データセットの作成および参照時における処置は DISP パラメータで指定できる。F4 システムでは, NEW, OLD, MOD, および SHR の 4 つの処置が指定できるが, それらの機能はデータセットの編成法, データセットが既存か否か, 作成か参照かによって異なってくる。表 6.1 はこれら全ての場合について, 処置の詳細をまとめたものである。なお, データセットの指定が“データセット名(メンバ名)”であれば区分編成, “データセット名”であれば順編成として処置する。

6.3.5 領域確保パラメータについて

データセットの領域確保に関する指示は, 下記の SPACE パラメータを用いて行える。また, 既存データセットに対しては使用時に一部を変更できる。

SPACE=▼([トラック数][, [増分][, ディレク

表 6.1 データセットの処置の詳細

データ セット 処置	区分編成 (ディスクのみ)				順編成 (ディスク・磁気テープ)			
	既 存		非 既 存		既 存		非 既 存	
	作 成	参 照	作 成	参 照	作 成	参 照	作 成	参 照
NEW	実行時エラーとなる。		データセットと指定メンバを新規作成する。	実行時エラーとなる。	実行時エラーとなる。		データセットを新規作成し、先頭から書き込む。	実行時エラーとなる。
OLD	指定メンバを追加作成する。ただし、指定メンバが存在すると置き換えられる。	指定メンバを読み込む。	データセットが見つからないため実行時エラーとなる。		既存データセットの先頭から書き込む。	指定データセットを読み込む。	データセットが見つからないため実行時エラーとなる。	
MOD	指定メンバを追加作成する。ただし、指定メンバが存在すると実行時エラーとなる。	指定メンバを読み込む。	データセットが見つからないため実行時エラーとなる。		既存データセットの後へ追加書きする。	指定データセットを読み込む。	データセットを作成し、先頭から書き込む。	データセットが見つからないため、実行時エラーとなる。
^(註) SHR	処置はOLDと同じであるが、SHR指定があると、実行時他のジョブと同時に指定データセットにアクセス可能となる。ただし、SHR指定により書き込みを行うジョブが同時に複数存在すると共にエラーとなるため、書き込み時にはSHR指定をしない方が望ましい。				(ディスクの場合) 処置はOLDと同じであるが、SHR指定があると実行時他のジョブと同時に指定データセットにアクセスができる。ただし、複数ジョブが存在する時、1つでも書き込みを行うジョブが存在すると、全てエラーとなる。 (磁気テープの場合) 処置はOLDと同じである。磁気テープの場合、1つの媒体を共用することはできないが、共にSHR指定があれば複数ジョブで同一名のデータセットを扱うことがあっても、ジョブ処理が待たされることはない。			

(注) データ管理機能の処置は上記の通りであるが、JES/Eのセットアップ機能を積極的に有効利用する観点からすれば、データセットの参照時にはできるだけDISP=SHRとし、データセットの作成時には決してSHRを指定してはならない。⁽¹²⁾

トリブロック数]) [, RLSE] ▼

ここで、右辺の各サブパラメータの意味は次の通りである。

(1) トラック数

データセット作成時に確保するトラック数を指定する。なお、区分編成データセットの場合にはディレクトリ領域もこの中に含める。

(2) 増 分

データセット作成中、または拡張中に領域が不足した場合に、その都度拡張するトラック数を指定する。システムは拡張処理時にできるだけ連続領域を割当てが、連続割当てができない場合には最大 5 個のエクステンツまでの処理を行う。また増分指定時の領域拡張処理は最大 16 回まで行われる。増分値を省略すると領域が不足した時点でエラーとなる。

(3) ディレクトリブロック数

区分データセット作成時には必ず指定しなければならない。なお、1 ディレクトリブロック当り 256 バイトの領域が確保されるが、この中にソースモジュール、オブジェクトモジュールならば 20 個程度、ロードモジュールならば 7 個程度のメンバが格納できる。しかし TSS の特殊機能により作成したメンバの場合、5 個程度しか格納できないので、少し余裕をもってディレクトリブロック数を指定することが望ましい。

(4) RLSE

データセットのクローズ時に、確保した領域のうちの未使用領域を解放することを指示するサブパラメータであり、作成時、使用時共に指定できる。

(5) 記述法

データセット作成時には全てのサブパラメータが指定でき、順編成の場合には

SPACE = ▼ (トラック数 [, 増分])
[, RLSE] ▼

区分編成の場合には

SPACE = ▼ (トラック数 , [増分] ,

ディレクトリブロック数)

[, RLSE] ▼

のように記述する、また既存データセット使用時には

SPACE = ▼ [(, 増分)] [, RLSE] ▼

のように記述しなければならない。なお、ユーザの便宜を計るため、データセット作成時には SPACE パラメータを全て省略可能なようにカタプロを設計するが、1 つでもサブパラメータを指定する必要がある、上記の通り記述しなければならない。

6.4 各カタプロの設計

6.4.1 JOB 文

本文はカタプロ呼出し文ではなく、F4 システムのジョブ文そのものである。

(1) 記述形式

表 6.2

(2) 機能上の特記事項

(2.1) 本文は F4 システムの JOB 文そのものであるが、課金識別情報記述用位置パラメータを利用して、ジョブクラスの記述法を簡単化し、かつ LP 出力打切行数とカード出力打切枚数の指定が可能なものにする。これら 3 つの位置パラメータは、表 6.2 の順序で記述しなければならない。また、表 6.2 以外の JOB 文パラメータはシステム運用値がとられ、ユーザが指定しても有効とはならない。

(2.2) ユーザ登録名^(注)を省略することはできない。

(2.3) F4 システムには LP 出力頁数を制限する機能がない。その対策として、JES/E 制御文の MAIN 文の LINES パラメータを利用して LP 出力打切行数が指定できるようにする。⁽¹²⁾このため、指定された行数は論理レコード数とみなされる。論理レコード数をユーザが計算する場合、LP の 1 行を 1 論理レコードと考えれば良いが、改頁、2 行改行などをして 1 行改行としか扱われず、逆に同行に重ね

(注) ユーザ登録名は 7 文字で表現し、形式は次の通りとする。

ユーザ登録名 ; A 01 PASS

① ② ③

ここで、①は部識別を、②は部内ユーザ番号を、③はパスワードを表わす。なお、先頭の 3 文字をユーザ名と呼ぶ。

打ちをした場合には改行と同じに扱われるので注意を要する。

(2.4) ジョブクラス、LP出力打切行数、カード出力打切行数を指定する場合、全体を活弧でくくらなければならないが、ジョブクラスのみ指定する場合には活弧を省略できる。

(2.5) システム運用上、ジョブの入力元に許すジョブの出力先は次表の如く制限する。

入力元 \ 出力先	オープン出力先(O)	クローズ出力先(C)	リモート出力先(R)	備考
オープン機器	◎	×	×	オープンバッチジョブ
クローズ機器	○	◎	×	クローズバッチジョブ
リモート端末	○	×	◎	リモートバッチジョブ
T S S 端末	◎	×	×	会話型リモートバッチジョブ

注) ◎は標準出力先, ○は指定可能出力先, ×は指定不可能出力先を表わす。

(2.6) ジョブステップの実行を行うか否かの判定に使用できる演算記号には以下のものがある。

EQ; 指定コードが完了コードと等しい。

NE; 指定コードが完了コードと等しくない。

GT; 指定コードは完了コードより大きい。

GE; 指定コードは完了コードより大きいまたは等しい。

LT; 指定コードは完了コードより小さい。

LE; 指定コードは完了コードより小さいまたは等しい。

条件は8個まで指定できる。条件が1つの場合には外側の活弧が省略できる。

(2.7) NOTIFYパラメータを指定すると、このジョブ終了後TSS端末に、ジョブ識別番号^(註)およびジョブ終了時刻が通知される。

(2.8) ユーザジョブ処理自動継続機能を利用してジョブを再開(RESTART)する場合には、ジョ

ブストリームの先頭のJOB文でRESTARTパラメータを指定し、2枚目に決められた記述のDD文を挿入する以外は、チェックポイントをとった時のジョブストリームのままでなければならない(使用例参照)。

(3) 使用例

(3.1) パラメータの指定を行わない。この場合、ジョブクラスはSHRTとみなされる。

```
//A01PASS JOB
```

(3.2) ジョブクラスはSHRTであるが、LPとカードの打切値を指定する。

```
//A01PASS JOB (,600,100)
```

(3.3) ジョブクラスをMIDLとする。

```
//A01PASS JOB MIDL
```

(3.4) ジョブの打切時間を5分とし、ジョブ終了をTSS端末に通知させる。

```
//A01PASS JOB TIME=5,  
// NOTIFY=A01
```

(3.5) ジョブクラスをLONG、ジョブ打切時間を5分30秒としてジョブをリモート端末から投入し、その結果をオープン出力とする。

```
//A01PASS JOB LONG,  
// TIME=(5,30),  
// MSGCLASS=O
```

(3.6) ジョブの終了条件を指定する。

```
//A01PASS JOB COND=(0,NE) (i)  
//A01PASS JOB COND=((0,EQ),  
// (12,EQ),(24,EQ)) (ii)  
//A01PASS JOB COND=(16,LE) (iii)  
//A01PASS JOB COND=((4,GT),  
// (12,LT)) (iv)
```

(註) ジョブ識別番号は7文字で表現し、形式は次の通りとする。

ジョブ識別番号; A01R123

① ② ③

①はユーザ名, ②は入力元, ③は該当ユーザのジョブ追番を表わす。

- (i) 完了コードが0でなければジョブを終了する。
- (ii) 完了コードが0, 12, 24の場合ジョブを終了する。
- (iii) 完了コードが16より大きい場合ジョブを終了する。
- (iv) 完了コードが4～12以外の場合ジョブを終了する。

(3.7) ユーザジョブ処理自動継続機能を利用してジョブ処理を再開する。

```
//A01PASS JOB LONG,
      RESTART=(GO,GO,POINT)
//SYSCHK DD DSN=A01.CHECK,
//      DISP=OLD
(以下、チェックポイントをとった時と同一のジョブストリームとする)
```

銚 GO文のステップ名がGOである場合

6.4.2 APTRAN文

(1) 記述形式

表 6.3

(2) 機能上の特記事項

(2.1) APTRAN文の出力を個人ファイルに保存しない場合には、後続のFORTC文でTS=INの指定をすれば自動的にAPTRAN文の出力をそのFORTC文の入力としえる(使用例参照)。

(2.2) 制御データをカード入力する場合はdd名がFT01なるDD文を用意しなければならない(使用例参照)。

(2.3) APプログラムと制御データを共にカード入力する場合、前者を先に入力するとAPプログラムのカード入力を指示するDD文はいらないが、前者を後に入力する場合にはAPプログラムのカード入力を指示するための、dd名がSYSINなるDD文を用意しなければならない(使用例参照)。

(2.4) APTRAN文の出力形式とFORTC文の入力形式を統一するために、新規作成のデータセットには特有のレコード長、ブロック長、レコード形式を展開する。しかしDISP=NEW以外の場合にこの処置をすると既存データセットのレコード長等を変更してしまう恐れがある。このため、TFパラメータを指定してAPTRAN文の出力を個人ファ

イルに保存する場合、DISP=NEW以外でデータセットの新規作成は不可能として、矛盾の発生を防止する。

(3) 使用例

(3.1) APプログラム、制御データ共にカード入力し、その出力をFORTC文へ引渡す。

```
// EXEC APTRAN
//FT01 DD *
  制御データカード
//SYSIN DD *
  APソースプログラムカード
// EXEC FORTC,TS=IN
```

または

```
// EXEC APTRAN
  APソースプログラムカード
//FT01 DD *
  制御データカード
// EXEC FORTC,TS=IN
```

(3.2) 個人ファイル上のAPプログラム(順編成)を入力とし、出力を個人ファイル上に保存する。その際、領域指定は省略値を使用する。

```
// EXEC APTRAN,SF=▼A01.APU▼,
      TF=▼A01.CPU▼,ORG=PS
//FT01 DD *
  制御データカード
```

(3.3) 個人ファイル上のAPプログラム(区分編成)と制御データ(順編成)を入力して得られた出力と、APプログラムをカード入力し、制御データ(順編成)を個人ファイルから入力して得られた出力を連結してFORTC文へ引渡す。

```
// EXEC APTRAN,
//      SF=▼A01.APU1(MAIN)▼,
//      CF=▼A01.CONTPRG▼
// EXEC APTRAN,
//      CF=▼A01.CONTPRG1▼
  APソースプログラムカード
// EXEC FORTC,TS=IN
```

6.4.3 FORTC文

(1) 記述形式

表 6.4

(2) 機能上の特記事項^(注)

(2.1) FORTRAN77 コンパイラオプションを指定する PARAM パラメータに関しては第 5.3.3 項の (2.9) を参照のこと。

(2.2) SF パラメータは、SF, SF1, …, SF4 の順序で 5 個までデータセットが指定できる。複数のデータセットを指定する場合、第 5.3.12 項の (2.3) の条件を満たしていなければならない。また、SF パラメータを指定した時にはソースプログラムのカード入力はできない。

(2.3) ソースプログラムをカード入力する場合、それを指示する DD 文を用意する必要はないが、GO オプションを指定し、実行時使用データをカード入力する場合には、dd 名が FT05 なる DD 文を用意しなければならない(使用例参照)。

(2.4) SF パラメータで区分編成データセットを指定する場合、参照するメンバ名は SF パラメータと一緒に指定するか、または PARAM パラメータの ELM オプションで指定しなければならない。後者の場合、指定メンバ名が複数の指定データセットに存在すると、先に指定されたデータセットのメンバが有効となる。

(2.5) SF パラメータで指定したデータセットの全メンバを参照する場合には PARAM=▼ELM(*)▼ と記述する。

(2.6) COND パラメータの条件は、EVEN, ONLY を含めて 8 個まで指定できる。ただし、JOB 文でも COND パラメータの指定があると、本カタログ文の処理以前にジョブ終了となることがある。また、EVEN または ONLY を指定した場合、本カタログ文は条件パラメータ指定の有無と、先行カタログ文の終了状態とによって実行される場合とされない場合とがある。⁽⁴⁾ 下表参照。

COND オペランド		先行ジョブステップの状態	ジョブステップの実行
EVEN ONLY の指定	条件パラメータの指定		
指定なし	指定なし	すべて異常終了せず	実行される。
		どれかで異常終了	バイパスされる。
	指定あり	すべて異常終了せず	条件パラメータが一つでも満足されるとバイパスされる。
		どれかで異常終了	バイパスされる。
EVEN	指定なし	すべて異常終了せず	実行される。
		どれかで異常終了	実行される。
	指定あり	すべて異常終了せず	条件パラメータが一つでも満足されるとバイパスされる。
		どれかで異常終了	条件パラメータが一つでも満足されるとバイパスされる。
ONLY	指定なし	すべて異常終了せず	バイパスされる。
		どれかで異常終了	実行される。
	指定あり	すべて異常終了せず	バイパスされる。
		どれかで異常終了	条件パラメータが一つでも満足されるとバイパスされる。

(注) システムの運転操作上、SAVEHALT コマンドを用いなければならない場合には⁽¹³⁾ユーザによる FORTRAN77 の GO オプションの使用を禁止しなければならない。

(2.7) SINCパラメータで指定できるデータセットは区分編成に限られる。また、SINC, SINC1, ..., SINC4の順に最大5個のデータセットが指定できる。ただし、参照するメンバ名はユーザプログラムの中の*INCLUDE行で指定しなければならない。

(2.8) TS=INを指定すれば、個人ファイルに保存指定のない先行APTRAN文の出力をFORTC文の入力とすることができる。また、TSパラメータを指定した場合にはソースプログラムのカード入力はできない。

(2.9) コンパイラの出力はできるだけ保存しないこと(第3.2.1項参照)。やむを得ず保存する場合には、第5.3.3項の(2.10)を参照のこと。

(3) 使用例

(3.1) ソースプログラムをカード入力し、翻訳結果を印刷出力する。

```
// EXEC FORTC
```

```
ソースプログラムカード
```

(3.2) 個人ファイル上のソースプログラムを入力し、翻訳結果を印刷出力する。

```
// EXEC FORTC, PARAM=▼ELM(*)▼,
// SF=▼A01.MAIN▼ (i)
// EXEC FORTC, SF=▼A01.SUB1▼,
// SF1=▼A01.SUB2▼,
// SF2=▼A01.SUB3▼ (ii)
// EXEC FORTC,
// SF=▼A01.ABC(PRG1)▼,
// SF1=▼A01.DEF(PRG2)▼ (iii)
// EXEC FORTC,
// PARAM=▼ELM(PRG1, PRG2)▼,
// SF=▼A01.ABC▼,
// SF1=▼A01.DEF▼ (iv)
```

(i) 区分編成データセットの全メンバを入力する。

(ii) 順編成データセットを3個入力する。

(iii) SFパラメータで入力メンバ名も指定する。

(iv) 入力するメンバ名をELMオプションで指定する。この場合、A01.ABCおよびA01.DFFにメンバPRG2が共に存在すると、先に指定したデータセットのメンバが有効となり、A01.DEFのメンバ

PRG2は入力されない。

(3.3) GOオプションを指定し、かつ実行時使用データをカード入力する。

```
// EXEC FORTC, PARAM=GO
```

```
ソースプログラムカード
```

```
//FT05 DD *
```

```
入力データカード
```

(3.4) PARAMパラメータでGOとNOSOURCE(コンパイラオプション), MAP(ローダオプション)を指定する。

```
// EXEC FORTC,
```

```
// PARAM=▼GO, NOSOURCE /MAP▼
```

```
ソースプログラムカード
```

(3.5) PARAMパラメータでGO, MAP, DATAON=5(実行可能プログラムオプション)を指定し、実行時使用データをカード入力する。

```
// EXEC FORTC,
```

```
// PARAM=▼GO/MAP/DATAON=5▼,
```

```
// SF=▼A01.ABC(MAIN)▼
```

```
//FT05 DD *
```

```
入力データカード
```

(3.6) ユーザプログラムの中で*INCLUDE行を使用する。

```
// EXEC FORTC,
```

```
// SINC=▼A01.DATA▼
```

```
ソースプログラムカード
```

(3.7) 個人ファイル上のソースプログラム(順編成)を複数個入力して得られる出力と、ソースプログラムをカード入力して得られる出力を連結してLIED文へ引渡す。

```
// EXEC FORTC, SF=▼A01.SUB1▼,
```

```
// SF1=▼A01.SUB2▼,
```

```
// SF2=▼A01.SUB3▼
```

```
// EXEC FORTC
```

```
ソースプログラムカード
```

```
// EXEC LIED
```

6.4.4 LIED文

(1) 記述形式

表 6.5

(2) 機能上の特記事項

(2.1) 本文の出力を保存しなければ、LIED文の出力を自動的に後続GO文へ引渡すことができる。保存した場合でも本文にステップ名をつけ、かつGO文のSNパラメータでそのステップ名を指定すれば、LIED文の出力を自動的にそのGO文へ引渡すことができる。この場合、ステップ名の指定は必須となる(GO文参照)。

(2.2) 第5.3.5項の(2.1), (2.2), (2.3), (2.5)項参照のこと。

(2.3) ALIBパラメータでは、ALIB, ALIB-1, ..., ALIB4という順序で最大5個のデータセットの指定ができる。

(2.4) IEFパラメータでは、IEF, IEF1, ..., IEF4という順序で最大5個のデータセットの指定ができる。ただし、メンバ名はINCLUDE文で指定しなければならない。使用例参照。

(2.5) オブジェクトモジュールはできるだけ保存すべきでない(第3.2.1項参照)。やむを得ず個人ファイルに保存している場合には、RFパラメータでRF, RF1, ..., RF4という順序で最大5個のデータセットが指定できる。

(3) 使用例

(3.1) 先行のFORTC文の出力を受取りそれを結合編集する。

```
// EXEC LIED
```

(3.2) 先行カタプロ文との関係を断ち、個人ファイル上のロードモジュールを主要入力とする。

```
// EXEC LIED, S=ON,
// IEF=▼A01.EB1▼,
// IEF1=▼A01.EB2▼
INCLUDE IEF(MAIN), IEF1(SUB1)
```

(3.3) 先行カタプロ文との関係を断ち、個人ファイル上のロードモジュールを主要入力とし、かつユーザライブラリを自動呼出入口とする。

```
// EXEC LIED, S=ON,
// IEF=▼A01.EB▼,
// ALIB=▼A01.LIB1▼,
// ALIB1=▼A01.LIB2▼
INCLUDE IEF(MAIN)
```

(3.4) 先行のFORTC文の出力を受取り個人ファイル上のロードモジュールを追加入力して、出力を個人ファイル上に新規作成する。SPACEパラメータの値は省略値を用いる。

```
// EXEC LIED,
// IEF=▼A01.MASTER▼,
// OEF=▼A01.MASTER1(B)▼
INCLUDE IEF(A)
```

(3.5) 先行FORTC文の出力を受取り、結合編集して、出力を個人ファイル上の既存データセットに保存する。

```
// EXEC LIED,
// OEF=▼A01.EB(PRG1)▼,
// DISP=OLD
```

(3.6) 先行FORTC文の出力を受取り、結合編集して、出力を個人ファイル上に新規作成する。

```
// EXEC LIED,
// OEF=▼A01.EB1(PRG)▼,
// SPACE=▼(10,5,5)▼
```

(3.7) 1つのジョブストリームの中に複数のLIED文を挿入する。

```
⋮
// EXEC LIED
⋮
// EXEC LIED, R=ON
⋮
// EXEC LIED,
// OEF=▼A01.PGM(PART1)▼
⋮
// EXEC LIED, R=ON
⋮
```

(3.8) 先行FORTC文の出力を受取り、ユーザライブラリを自動呼出入口として結合編集し、出力

を個人ファイルに保存すると共に後続 GO 文へ引渡す。

```
//L EXEC LIED,
//      ALIB=▼A01.LIB▼,
//      OEF=▼A01.PGM(P1)▼
// EXEC GO, SN=L
```

6.4.5 GO 文

(1) 記述形式

表 6.6

(2) 機能上の特記事項

(2.1) GO 文で定義する実行時使用データをカードから入力する場合も個人ファイルから入力する場合も共にプログラム中の入力文のデータセット識別番号は 5 としなければならない。

(2.2) DF パラメータでは、DF, DF1, ..., DF4 という順序で最大 5 個のデータセットを指定できる。ただし、これらのデータセットは第 5.3.12 項の (2.3) の条件を満たしていなければならない。

(2.3) DF パラメータを指定し、かつデータカード入力も行う場合には、個人ファイル上のデータの後にカード入力のデータが付加される。

(2.4) 実行時使用データをカード入力する場合、それを指示する DD 文を用意する必要はない。

(2.5) TIME パラメータを JOB 文で指定している場合には、本文で指定した値に達しなくても JOB 文で指定した値を満たした時点でジョブ終了となる。

(2.6) RD パラメータを指定すれば、ユーザジョブ処理自動継続機能により、中断点におけるチェックポイントをとってからジョブ終了となる。このジョブは JOB 文の RESTART パラメータを指定すれば自動的に再開できる。ただし、ジョブストリームの記述は規約どおりでなければならない（使用例参照）。

(2.7) SN パラメータで本文の入力としたい LIED 文のステップ名を指定すれば、出力を保存する LIED 文の出力も自動的に GO 文の入力とし得る。なお、出力を保存しない LIED 文に対する SN パラメータの指定は無意味である。

(3) 使用例

(3.1) 実行時使用データをカード入力して、出

力保存の指定のない先行 LIED 文の出力を 2 度実行する。

```
// EXEC GO
//      入力データカード
// EXEC GO
//      入力データカード
```

(3.2) 実行形式プログラムを個人ファイルから入力する。

```
// EXEC GO, EF=▼A01.PGM▼,
//      EMEM=NO1
```

(3.3) 出力を保存する指定のある先行 LIED 文の出力を 2 度実行する。

```
//L EXEC LIED,
//      OEF=▼A01.P(PGM3)▼
// EXEC GO, SN=L
//      入力データカード
// EXEC GO, SN=L
//      入力データカード
```

または、

```
// EXEC LIED,
//      OEF=▼A01.P(PGM3)▼
// EXEC GO, EF=▼A01.P▼,
//      EMEM=PGM3
//      入力データカード
// EXEC GO, EF=▼A01.P▼,
//      EMEM=PGM3
//      入力データカード
```

(3.4) FORTC 文、LIED 文の完了コードにより、GO 文の実行条件を定める。

```
//F EXEC FORTC, ...
//L EXEC LIED, ...
// EXEC GO, COND=((8,GE,F),
//      (4,GE,L))
//      入力データカード
```

(3.5) ユーザジョブ処理自動継続機能を利用してチェックポイントをとってからジョブを終了する。

```

:
//GO EXEC GO, RD=NR
//CHECK DD DSN= A01.CHECK,
//    DISP=NEW,
//    SPACE=(TRK,(5,2),RLSE)
:

```

6.4.6 LINKGO文

(1) 記述形式

表 6.7

(2) 機能上の特記事項^(註)

(2.1) EF (または RF) パラメータを指定すると、自動的に先行カタプロ文との関連が断たれる。

(2.2) 第 5.3.7 項の (2.2), (2.3), (2.4), (2.5) を参照のこと。

(2.3) 実行時使用データをカード入力する場合、それを指示する DD 文を用意する必要はない。

(2.4) EF, RF パラメータを共に指定した場合には、RF パラメータは無効となる。

(3) 使用例

(3.1) 先行の FORTC 文の出力を受取り、ユーザライブラリを自動呼出入力として結合編集し実行する。なお、実行時使用データはカード入力する。

```

// EXEC LINKGO,
//    ALIB=▼A01.LIB1▼

```

入力データカード

(3.2) 先行カタプロ文との関連を断ち、個人ファイル上のロードモジュールを主要入力として結合編集し実行する。なお、実行時使用データは個人ファイルから入力する。

```

// EXEC LINKGO,
//    EF=▼A01.A(MAIN)▼,
//    EF1=▼A01.A(SUB1)▼,
//    EF2=▼A01.A(SUB2)▼,
//    DF=▼A01.DATA▼

```

6.4.7 ULIB文

(1) 記述形式

表 6.8

(2) 機能上の特記事項

(2.1) コンパイラの NAME オプション、リンケージディタの NCAL オプションは自動的に指定されるため、FPARAM で NAME, NONAME オプション、LPARAM で NCAL オプションを指定してはならない。

(2.2) EF パラメータで指定したデータセット中に、作成されるメンバと同一のメンバ名が存在すると DISP=OLD 指定時には置換えとなり、DISP=MOD 指定時にはエラーとなる。また、同一のメンバ名が存在しない場合には、どちらの場合も追加作成される。

(2.3) 第 5.3.8 項の (2.4), (2.5), (2.6) を参照のこと。

(3) 使用例

(3.1) ソースプログラムをカード入力して、個人ファイル上にユーザライブラリを作成する。

```

// EXEC ULIB, EF=▼A01.SUB▼,
//    SPACE=▼(10,5,5)▼

```

ソースプログラムカード

(3.2) ソースプログラムを個人ファイルから入力して、個人ファイル上にユーザライブラリを作成する。なお SPACE パラメータの値は省略値を用いる。

```

// EXEC ULIB, SF=▼A01.A(SUB1)▼,
//    SF1=▼A01.A(SUB2)▼,
//    SF2=▼A01.A(SUB3)▼,
//    EF=▼A01.ULIB▼

```

6.4.8 USDK文

(1) 記述形式

表 6.9

(2) 機能上の特記事項

(2.1) 本文は実行を伴うカタプロ文 (GO オプション指定のある FORTC 文, GO 文, LINKGO 文) の後に挿入しなければならない。

(2.2) BSIZE, RSIZE, RECFM パラメータは新規作成のデータセットの場合のみ必要であるため、DISP=NEW 指定時のみユーザの指定が有効となる。データセット作成時にこれらのパラメータの指定がない場合には、システムの標準値 (BSI-

^(註) システムの運転操作上、SAVEHALT コマンドを用いなければならない場合には、⁽¹³⁾ユーザによる LINKGO 文の使用を禁止しなければならない。

ZE = 800, RSIZE = 800, RECFM = U) となる。

(2.3) ACCパラメータは参照のみが許されているデータセットを使用する場合には省略できない。このパラメータ指定時にはFILEパラメータで指定したデータセットに書き込みはできない。

(2.4) FILEパラメータは、FILE, FILE1, ..., FILE4 という順序で最大 5 個のデータセットが指定できる。ただし書き込みを行う場合には 2 番目以降のデータセットの指定は意味をもたない。

(2.5) DUMMYパラメータを指定した時には、RNO以外のパラメータは無効となる。

(2.6) DUMMYパラメータを指定しない時には、FILEパラメータを省略できない。

(2.7) RNOパラメータの値は、00 ~ 04, 08 ~ 97 なる 2 桁の数字を指定する。

(2.8) GO 文の出力結果を、X8 システムの個人ファイルに HICS⁽¹⁰⁾ でファイル伝送することを予定している場合には、本文を使用して F4 システムの個人ファイルに格納しておかなければならない。その際、RSIZE, BSIZE および RECFM は送り先ファイルの値に合せておかなければならない。但し、RECFM では不定長形式を指定してはならない。

(3) 使用例

(3.1) 実行時に使用するデータセット識別番号 1 のデータセットをダミーとする。

```
// EXEC GO, EF=▼A01.A▼,
//     EMEM=PGM
// EXPAND USDK, RNO=01,
//     DUMMY=ON
```

(3.2) FORTC 文の GO オプション実行時の出力データを、新規作成の順編成データセットにデータセット識別番号 2 で書き込む。なお、SPACE, BSIZE, RSIZE, RECFM パラメータの値は省略値を用いる。

```
// EXEC FORTC, PARAM=GO,
//     SF=▼A01.B(PGM)▼
// EXPAND USDK, RNO=02,
//     FILE=▼A01.DATA▼,
//     DISP=NEW, ORG=PS
```

(3.3) GO 文実行時の出力データを、データセット識別番号 10 で新規作成の順編成データセットに書き込み、後続の GO 文実行時にデータセット識別番号 11 で参照する。

```
// EXEC GO, EF=▼A01.A▼,
//     EMEM=PGM1
// EXPAND USDK, RNO=10,
//     FILE=▼A01.B▼, DISP=NEW,
//     SPACE=▼(5,2)▼,
//     BSIZE=800, RSIZE=80,
//     RECFM=FB
// EXEC GO, EF=▼A01.A▼,
//     EMEM=PGM2
// EXPAND USDK, RNO=11,
//     FILE=▼A01.B▼
```

6.4.9 USWK 文

(1) 記述形式

表 6.10

(2) 機能上の特記事項

(2.1) 第 6.4.8 項の (2.1), (2.2), (2.5), (2.7) を参照のこと。

(2.2) USWK 文では、順編成データセットのみを取扱う。

(2.3) データセット名はピリオド等の特殊文字を含まない 8 文字以内の英数字でなければならない。ただし、後続ジョブステップで使用しない場合には省略できる。

(2.4) ジョブ処理中に USWK 文で作成されたデータセットは、ジョブ終了と共に全て消去される。

(3) 使用例

(3.1) GO 文実行時にデータセット識別番号 3 で使用するユーザ作業用データセットをダミーとする。

```
// EXEC GO, EF=▼A01.A▼,
//     EMEM=B
// EXPAND USWK, RNO=03,
//     DUMMY=ON
```

(3.2) FORTC 文の GO オプションによる実行時に、データセット識別番号 8 でユーザ作業用データセットを作成する。なお SPACE, BSIZE, RSIZE, RECFM パラメータの値は省略値を用いる。さらに、このデータセットは後続の GO 文実行時にデータセット識別番号 3 で参照する。

```
// EXEC FORTC, PARAM=GO,
// SF=▼A01.SORT(A)▼
// EXPAND USWK, RNO=08,
// FILE=WORK
// EXEC GO, EF=▼A01.P▼, EMEM=Q
// EXPAND USWK, RNO=03,
// FILE=WORK, DISP=OLD
```

6.4.10 CF 文

本文では、オープンバッチジョブストリームでは使用できない磁気テープに関するパラメータが含まれているので注意を要する。磁気テープを使用する場合には申請ジョブとしなければならない。

(1) 記述形式

表 6.11

(2) 機能上の特記事項

(2.1) オープンバッチジョブストリームでは、UNIT=MT を指定できない。したがって、オープンバッチジョブストリームで CF 文を使用する場合、FILE パラメータ以外は全て無効である。

(2.2) 本文を用いて入力データセットを連結する場合には、必ず親となる USDK 文または USMT 文が先行していなければならない。

(2.3) 連結する全てのデータセットは、第 5.3.12 項の (2.3) の条件を満たしていなければならない。

(2.4) カード入力データを連結する場合、カード入力を指示するための DD 文を用意しなければならない（使用例参照）。

(2.5) CF 文で複数の FILE パラメータは使用

できない。

(2.6) UNIT=MT を指定した CF 文を使用する場合には、申請ジョブとしなければならない。

(2.7) 磁気テープを使用する場合、VOL パラメータは省略できない。このため NL 形式の場合にもボリューム通番を使用本数だけ指定しなければならない。この時ボリューム通番は任意で良い。なお、VOL パラメータで指定できるボリューム通番の数は 255 個までである。

(2.8) SL 形式の場合、指定データセット順序番号の位置に指定データセットがないとエラーとなる。

(3) 使用例

(3.1) オープンバッチジョブストリームでの使用例。

- USDK 文で定義したデータセットに、個人ファイル上のデータセットとカード入力データとを連結する。

```
// EXPAND USDK, RNO=03,
// FILE=▼A01.DATA1▼
// EXPAND CF, FILE=▼A01.DATA2▼
// DD *
```

入力データカード

上の例は次のように記述しても良い。

```
// EXPAND USDK, RNO=03,
// FILE=▼A01.DATA1▼,
// FILE1=▼A01.DATA2▼,
// DD *
```

入力データカード

(3.2) 申請ジョブストリームでの使用例。

- USMT 文で定義したデータセットに、個人ファイル上のデータセットとカード入力データとを連結する。なお USMT 文については第 6.4.29 項を参照のこと。

註 F4 システムでは新しくデータセット順序番号という必須パラメータを設けて、磁気テープデータのシステム処理効率を一段と向上させている。さらに、データセット名と照合する機能も設けて磁気テープデータの保護を計っている。データセット順序番号の指定は一見不便そうに思えるが、データセット順序番号の照会ユーザコンソールで容易に行えるようにして、むしろ積極的にデータセット順序番号を使用すべきである。

```
// EXPAND USMT, RNO= 02,
//   FILE=DATA1, VOL=NL0001
// EXPAND CF,
//   FILE=▼A01.DATA2▼
// DD *
```

入力データカード

- USDK文で定義したデータセットに、磁気テープ上のデータとカード入力のデータ、および個人ファイル上のデータセットとを連結する。

```
// EXPAND USDK, RNO= 03,
//   FILE=▼A01.XYDATA▼
// EXPAND CF, UNIT=MT,
//   FILE=XYDATA1, FSEQ= 3,
//   VOL=NM0111
// DD *
```

入力データカード

```
// EXPAND CF,
//   FILE=▼A01.XYDATA3▼
```

6.4.11 MTR文

(1) 記述形式

表 6.12

(2) 機能上の特記事項

(2.1) MTR文でMT入力スプールされた実行時データは、それを参照するジョブステップが終了すると共に消去される。このため、同一の磁気テープデータを複数のジョブステップで参照する場合には、各カタプロ文の後に同一のMTR文を挿入しなければならない。

(2.2) ブロック長が26KB以上のデータセットを使用する場合には自動的にトラックオーバーフロー機能が指定される。ただし、この機能が指定されると、バックスペース機能を使用することができなくなるので注意を要する。

(2.3) 第6.4.8項の(2.7)と第6.4.10項の(2.7)と(2.8)を参照のこと。

(3) 使用例

(3.1) 先行LIED文から引渡された出力をGO文で実行する際にデータセット識別番号1で参照するSL形式の磁気テープデータを、ジョブ投入時に

MT入力スプールする。

```
// EXEC GO
// EXPAND MTR, RNO= 01,
//   FILE=▼AB.DATA▼, FSEQ= 3,
//   VOL=NL0123
```

(3.2) GO文で実行するプログラムが実行時にデータセット識別番号9で参照するNL形式の磁気テープデータを、ジョブ投入時にMT入力スプールする。なおデータはボリュームの先頭に存在し、3ボリュームにまたがっている。

```
// EXEC GO, EF=▼A01.XYPRG▼,
//   EMEM=MAH
// EXPAND MTR, RNO= 09,
//   VOL=▼A,B,C▼, BSIZE=800,
//   RSIZE= 80, RECFM=FB
```

(3.3) ジョブストリーム中の2つのGO文実行時に、同一の磁気テープデータを同一データセット識別番号10で参照する。なおデータセットはSL形式のボリュームの先頭に存在する。

```
// EXEC GO, EF=▼A01.MTJOB▼,
//   EMEM=ONE
// EXPAND MTR, RNO= 10,
//   FILE=DATA, VOL=NL0185
// EXEC GO, EF=▼A01.MTJOB▼,
//   EMEM=TWO
// EXPAND MTR, RNO= 10,
//   FILE=DATA, VOL=NL0185
```

6.4.12 MTW文

(1) 記述形式

表 6.13

(2) 機能上の特記事項

(2.1) MTW文によりディスク上に出力されたデータセットは同一ジョブステップ内では参照可能であるが、他のジョブステップでは参照できない。

(2.2) MT出力は磁気テープ上の指定されたデータセット順序番号の位置に書込まれる。追加書きはできない。追加書きを必要とする場合にはMTRW文を使用する。

(2.3) MTW文はSL形式の磁気テープしか扱わない。

(2.4) VOLパラメータでボリューム通番を1つ指定するだけで、最大5本のボリュームへの出力が可能である。

(2.5) 第6.4.11項の(2.2)と(2.3)を参照のこと。

(3) 使用例

(3.1) 先行LIED文から引渡された出力をGO文で実行する際にデータセット識別番号3で書込むSL形式の磁気テープデータをMT出力スプールする。なお、MT出力はボリューム通番NS0111の2番目のデータセットとして書込む。

```
// EXEC GO
// EXPAND MTW, RNO=03,
//   FILE=XYDATA, FSEQ=2,
//   VOL=NS0111
```

(3.2) GO文の実行時にデータセット識別番号2で書込むSL形式の磁気テープデータをMT出力スプールする。なお、MT出力はボリューム通番NS1111の1番目のデータセットとして書込む。

```
// EXEC GO, EF=▼A01.AB▼,
//   EMEM=B
// EXPAND MTW, RNO=02,
//   FILE=XY, VOL=NS1111
```

6.4.13 MTRW文

本文は実行時に磁気テープ上のデータセットを参照・更新する設計になっているユーザプログラムに対してもMTスプール機能が利用し得るようにするために設けたものである。本文を用意すればジョブ投入時に磁気テープデータはMT入力スプールされる。実行時にそのデータセットは参照・更新され、ジョブ終了後ユーザによる更新データの取出しまでシステム上に保管される。

(1) 記述形式

表 6.14

(2) 機能上の特記事項

(2.1) 本文で更新されたデータセットは実行したジョブステップ以外参照できない。

(2.2) MT出力はMT入力スプール時に読んだデータセットの位置へ上書きされる。

(2.3) 第6.4.11項の(2)参照のこと。

(3) 使用例

(3.1) 先行LIED文から引渡された出力をGO文で実行する際にデータセット識別番号10で参照・更新するSL形式の磁気テープデータをMT入力スプールし、ジョブ終了後MT出力する。

・最初のアクセスが指定データセットの読み込みの場合

```
// EXEC GO
// EXPAND MTRW, RNO=10,
//   FILE=MTDATA, FSEQ=3,
//   DISP=OLD, VOL=NM0135
```

・最初のアクセスが指定データセットへの追加書込みの場合

```
// EXEC GO
// EXPAND MTRW, RNO=10,
//   FILE=MTDATA, FSEQ=3,
//   VOL=NM0135
```

(3.2) GO文の実行時にデータセット識別番号11で参照・更新するNL形式の磁気テープデータをMT入力スプールし、ジョブ終了後MT出力する。

・最初のアクセスが指定データセットの読み込みの場合

```
// EXEC GO, EF=▼A01.MTPRG▼,
//   EMEM=B
// EXPAND MTRW, RNO=11,
//   DISP=OLD, VOL=▼A,B▼,
//   BSIZE=800, RSIZE=80,
//   RECFM=FB
```

6.4.14 CP文

(1) 記述形式

表 6.15

(2) 機能上の特記事項

(2.1) CP出力を行う場合、プログラムでは出力文のデータセット識別番号を7とすること。

(2.2) 本文は、CP出力を行うカタプロ文の後に挿入しなければならない。

(3) 使用例

(3.1) GO 文でプログラムのデバッグのみを行い、実際の CP 出力は行わない。

```
// EXEC GO, EF=▼A01.CPJOB▼,  
//      EMEM= CP1  
// EXPAND CP, DUMMY=ON
```

(3.2) FORTC 文の GO オプションで実行し、結果を CP 出力する。

```
// EXEC FORTC, PARAM=GO  
//      ソースプログラムカード  
// EXPAND CP
```

6.4.15 PP 文

(1) 記述形式

表 6.16

(2) 機能上の特記事項

(2.1) ユーザプログラム上ではプロッタの機種を意識せずにプロッタ共通のベシックソフトウェア (PSP) を使用でき、ジョブストリームの中でプロッタ出力を伴う実行文の後に本文を挿入すれば、実行結果はプリンタプロッタへの出力となる。

(2.2) 本文は 1 ジョブ当たり 1 枚に限られる。

(2.3) ユーザプログラムではプリンタプロッタ出力のためのデータセット識別番号を意識する必要がない。データセット識別番号は PLOT サブルーチンの中で自動的に 99 が発行される。

(3) 使用例

(3.1) プログラムデバッグのみを行い、実際にプリンタプロッタ出力は行わない。

```
// EXPAND PP, DEBUG=ON
```

(3.2) GO 文実行時にプリンタプロッタ出力を行う。

```
// EXEC GO, EF=▼A01.PPJOB▼,  
//      EMEM= P2  
// EXPAND PP
```

6.4.16 XY 文

(1) 記述形式

表 6.17

(2) 機能上の特記事項

(2.1) F4 システムのプロッタを使用する場合には、ユーザプログラム上でプロッタの機種を意識することなくプロッタ共通の PSP が使用でき、ジョブストリームの中でプロッタ出力を伴う実行文の後に本文を挿入すれば、実行結果は XY プロッタへの出力となる。ただし本文は 1 ジョブ当たり 1 枚に限る。また、ユーザプログラムでは XY 出力のためのデータセット識別番号を意識する必要がない。データセット識別番号は PLOT サブルーチンの中で自動的に 99 が発行される。なお、この場合本文の DEBUG, SETNO パラメータ以外は無効となる。

(2.2) F4 システムでプロッタ出力を伴うプログラムを実行するが、出力装置としては X8 システムの XY プロッタを使用する場合には XY 文のみ挿入可能とするが、次の事項に注意しなければならない。^(註)

(i) X8 システムの XY プロッタを使用する場合には、プロッタ出力を伴うプログラムの結合編集を行う文 (GO オプション指定のある FORTC 文, LI-ED 文, または LINKGO 文) で必ず PSP = ON を指定しなければならない。F4 システムでは PSP = ON の指定があれば X8 システム用のプロッタの PSP を自動的にユーザプログラムに結合する。その指定がなければ F4 システム用のプロッタの PSP を結合する。PSP = ON 指定で実行したプロッタ出力は F4 システム用プロッタに書出すことができない。逆に、PSP = ON 指定なしで実行したプロッタ出力は X8 システム用プロッタに書出せない。

(ii) PSP = ON 指定で実行するプログラムでは、データセット識別番号を PLOTS サブルーチンの引数で指定しなければならない。

(iii) JOB 文の MSGCLASS の指定にはよらず、データセット識別番号の引数を 99 とし、かつ本文

(註) F4, X8 システム共に同一メーカーの XY プロッタであれば (2.2) の記述は不必要となる。しかし、両システムの規模には圧倒的な相違があるので、XY プロッタの機種も自ずから異なるであろう。この場合、各メーカーの PSP によるプロッタ出力のデータ形式の間には互換性がないため、PSP は機種毎に独自のものを用意しなければならない。

でXYP=ONを指定すれば、F4システムでのジョブ処理終了後、システムが自動的にプロッタ出力をX8システムへ伝送してX8システムの出力待ちキューに登録し、ユーザによるXYプロッタへの取出しを待つ。なお、この場合本文のDEBUG, XYPパラメータ以外は無効となる。

(IV) また、データセット識別番号の引数を99とし、かつ本文のXYFパラメータで保存先データセット名を指定すれば、一旦プロッタ出力をF4システムの個人ファイルに格納することができる。このプロッタ出力をX8システム側からHICSコマンドを用いて、⁽¹⁰⁾F4システムの個人ファイルからX8システムの個人ファイルへ伝送すれば、X8システムのXYプロッタに書出すことができる。なお、この場合本文のDEBUG, XYF, DISPパラメータ以外は無効となる。

(V) 勿論、プロッタ出力を個人ファイルに保存する場合、本文ではなくUSDK文を使用しても良い。この場合、データセット識別番号として、00～04, 08～99の任意の値が使用できる(USDK文参照)。

(VI) XYPとXYFを共に指定した場合にはXYPのみが有効となる。

(3) 使用例

(3.1) プログラムのディバッグのみを行い、実際にXYプロッタ出力は行わない。

```
// EXPAND XY, DEBUG=ON
```

(3.2) 25 cm 幅の目盛付きの用紙にインクペンで書くことをオペレータに知らせる。

```
// EXPAND XY, SETNO=4
```

(3.3) F4システムで実行したプロッタ出力を、自動的にX8システムに伝送させ、X8システムのXYプロッタに書出す。なお、リストはセンタ出力とする。

```
//H01PASS JOB SHRT,  
// MSGCLASS=O  
// EXEC FORTC, SF=▼H01.A(B)▼,  
// PARAM=GO, PSP=ON  
// EXPAND XY, XYP=ON  
//
```

(3.4) F4システムでジョブを実行し、XYプロッタ出力にはX8システムのXYプロッタを使用するが、一旦F4システムの個人ファイルの既存データセットに格納しておく。

```
//H01PASS JOB SHRT  
// EXEC FORTC, SF=▼H01.A(B)▼,  
// PARAM=GO, PSP=ON  
// EXPAND XY, DISP=OLD  
// XYF=▼H01.XYOUT▼  
//
```

6.4.17 PPXY文

(1) 記述形式

表 6.18

(2) 機能上の特記事項

(2.1) ユーザプログラム上ではプロッタの機種を意識することなくプロッタ共通のPSPが使用でき、ジョブストリームの中でプロッタ出力を伴う実行文の後に本文を挿入すれば、実行結果はまずプリンタプロッタへの出力ができ、その後XYプロッタへの出力もできる。

(2.2) 本文は1ジョブ当り1枚に限る。

(2.3) ユーザプログラムではプリンタプロッタおよびXYプロッタ出力のためのデータセット識別番号を意識する必要がない。データセット識別番号はPLOTサブルーチンの中で自動的に99が発行される。

(2.4) 本文をできるだけ使用して、XY出力の前に出力結果のモニタを行うこと。不必要な結果はユーザコンソールから直ちに消去できる。

(3) 使用例

プロッタ出力を、XYプロッタに出力する前にプリンタプロッタでモニタするが、XYプロッタ出力時には25 cm 幅の白紙にインクペンで書くことをオ

ペレータに知らせる。

```
// EXPAND PPXY, SETNO=2
```

6.4.18 UPDATE 文

(1) 記述形式

表 6.19

(2) 機能上の特記事項

(2.1) データセットの編集作業は TSS 端末から行い、できるだけ本文を用いたバッチ処理に依るべきではない。

(2.2) 入力データはカードイメージの書式付きデータまたはソースプログラムに限る。

(2.3) OF, NF で指定したデータセットは、レコード長が 80 バイトの固定長でなければならない。なお、ブロック化係数は異っても良い。

(2.4) 次の機能制御文を使用すれば、順または区分編成データセットの各種編集作業が可能である（使用例参照）。

(i) ADD 制御文は作成および全体的順序番号付けの時使用する。

(ii) REPL 制御文は更新および全体的順序番号付けの時使用する。

(iii) REPRO 制御文は複写および全体的順序番号付けの時使用する。

(iv) CHANGE 制御文はカード単位の更新および部分的・全体的順序番号付けのとき使用する。

(2.5) 順編成データセットの更新を行う場合には、NF パラメータの指定をせずに、ORG=PS の指定をしなければならない。この場合、OF パラメータと NF パラメータで同一データセット名を指定してもエラーとなる。

(2.6) DISP=NEW の場合には装置の効率を考えたレコード長、ブロック長、レコード形式を展開するが、それ以外の場合には既存データセットのこれらの情報を変更する恐れがあるため展開しない。したがって、DISP=MOD で順編成データセットを新規作成するとレコード長等にシステムの標準値が入るため装置の効率が悪くなるので注意を要する。なお、この場合には、レコード長等の指定がないため、ワーニングメッセージが出る。

(2.7) OF または NF でメンバ名を指定し、か

つ機能制御文でもメンバ名を指定した場合には、後者のメンバ名が有効となる。ただし、複数のメンバ名を指定する場合には機能制御文で指定しなければならない。

(2.8) 本文に対する機能制御文およびデータをカード入力することを指示するために、INPUT なる dd 名の DD 文を必ず 1 枚用意しなければならない（使用例参照）。

(3) 使用例

(3.1) 順編成のデータセットを新規作成する。なお、SPACE パラメータの値は省略値を用いる。

```
// EXEC UPDATE, ORG=PS,
      NF=▼A01.SOURCE▼
//INPUT DD *
./ ADD [LIST=NO] (i)
[./ NUMBER NEW1=順序番号初期値,
  INCR=増分値] (ii)

```

(i) ADD 制御文は必須である。なお LIST パラメータを省略すると使用した制御文および入力データの全リストが出力される。リストが不要の場合には LIST=NO を指定する。

(ii) NUMBER 制御文は作成したデータセットに順序番号を付加したい場合のみ挿入する。

(3.2) 区分編成のデータセットを新規作成し、カード入力データを 2 つのメンバに保存する。

```
// EXEC UPDATE,
      NF=▼A01.SOURCE1▼,
      SPACE=▼(10,10,5)▼
//INPUT DD *
./ ADD NAME=A[,LIST=NO] (i)
[./ NUMBER NEW1=順序番号初期値,
  ./ INCR=増分値] (ii)

./ ADD NAME=B[,LIST=NO] (i)
[./ NUMBER NEW1=順序番号初期値,
  ./ INCR=増分値] (ii)

```

(i) ADD制御文は作成するメンバ毎に必須である。NAMEパラメータで保存先メンバ名を指定する。

(ii) NUMBER制御文は順序番号を付加したいメンバのみに挿入する。

(3.3) 入力データの内容を既存の順編成データセットへ登録する。順序番号の付加は行わない。

```
// EXEC UPDATE, DISP=OLD,
//      NF=▼A01.SOURCE▼
//INPUT DD *
./ ADD
  入力データカード
```

なお、上例においてDISP=MODを指定すると、データセットの内容追加の指示となる。

(3.4) 既存の区分編成データセットの中に、カード入力データを新規メンバとして保存する。

```
// EXEC UPDATE, DISP=OLD,
//      NF=▼A01.SOURCE1(B)▼
//INPUT DD *
./ ADD
  入力データカード
```

(3.5) 既存の区分編成データセットのメンバの内容をカード入力データの内容に置換え、それに順序番号を付加する。

```
// EXEC UPDATE, DISP=OLD,
//      NF=▼A01.PODATA▼
//INPUT DD *
./ REPL NAME=MEM1[,LIST=NO]
                                     (i)
./ NUMBER NEW1=順序番号初期値,
./      INCR=増分値
  入力データカード
```

(i) REPL制御文は指定メンバの内容の置換えを指示する文である。

(3.6) 順編成データセットの内容を、順編成データセットを新規作成して複写する。

```
// EXEC UPDATE, OF=▼A01.A▼,
//      NF=▼A01.B▼,
//      SPACE=▼(20,5)▼
//INPUT DD *
./ REPRO [LIST=ALL]
                                     (i)
```

(i) REPRO制御文は複写を示す文である。なお、LISTパラメータを省略すると、制御文のリストしか出力されない。複写データセットのリストが必要な場合はLIST=ALLを指定する。

(3.7) 区分編成データセットのメンバの内容を、既存の区分編成データセットに複写する。

```
// EXEC UPDATE, DISP=OLD,
//      OF=▼A01.MASTER▼,
//      NF=▼A01.COPY▼
//INPUT DD *
./ REPRO NAME=A1, RENAME=B1,
./      LIST=ALL
                                     (i)
./ REPRO NAME=A2
                                     (ii)
./ NUMBER NEW1=順序番号初期値,
./      INCR=増分値
                                     (iii)
```

(i) NAMEパラメータで複写するメンバ名を指定し、RENAMEパラメータで複写先のメンバ名を指定する。

(ii) RENAMEパラメータを省略すると、複写するメンバ名と同一メンバ名を指定したものとみなされる。

(iii) 複写する時、順序番号を付加したいメンバがあれば、そのメンバを指定したREPRO制御文の後にNUMBER制御文を挿入する。

(3.8) 順編成データセットの内容を、カード単位で修正、削除、追加を行う。

```
// EXEC UPDATE,
//      OF=▼A01.PSDATA▼, ORG=PS,
//      DISP=OLD
//INPUT DD *
./ CHANGE LIST=ALL
                                     (i)
  順序番号付の更新データカード
                                     (ii)
```



```

./ NUMBER INSERT=YES,
./ SEQ1 = 順序番号
./ NEW = 順序番号初期値,
./ INCR = 増分値 (iii)
  順序番号なしの挿入データカード
./ DELETE SEQ1 = 順序番号,
./ SEQ2 = 順序番号 (iv)

```

(i) CHANGE 制御文はカード単位でデータの更新を行うことを示す文である。LISTパラメータ省略時には、使用した制御文および入力データの全リストが出力される。修正された順編成データセットまたは区分データセットのメンバリストが必要な場合にはLIST=ALLを指定する。

(ii) 順序番号付きの更新データは、既存データセットの該当部分に挿入される。なお、同一順序番号のデータカードがあると置換えとなる。

(iii) INSERTパラメータで挿入データがあることを指示し、SEQ1パラメータでそのデータの挿入位置を指定する。なお、挿入データに順序番号を付加する場合にはNEW1とINCRパラメータで指定する。また、INSERTパラメータ省略時には順序番号のつけ直しとみなされる。

(iv) 順序番号の付いているデータはDELETE制御文で削除することができる。削除するデータの範囲はSEQ1, SEQ2 (SEQ1 < SEQ2) で指定する。1行だけ削除する場合はSEQ1パラメータのみを指定すれば良い。

(3.9) 区分データセットの内容をカード単位で修正、追加、削除を行う。

```

// EXEC UPDATE. DISP=OLD,
// OF=▼A01.FORTDATA▼
//INPUT DD *
./ CHANGE NAME=A, LIST=ALL (i)
  制御文およびデータカード
./ CHANGE NAME=B, RENAME=C,
./ LIST=ALL (ii)
  制御文およびデータカード

```

(i) NAMEパラメータで指定したメンバの内容を更新することを示す文である。

(ii) NAMEパラメータで指定したメンバの内容を更新し、かつ、更新データをRENAMEパラメータで指定したメンバに保存することを示す文である。

上例におけるCHANGE制御文の後に置く制御文およびデータカードは、(3.8)の例の(ii), (iii), (iv)と同じである。

(3.10) 入力側と出力側のデータセットの編成が異なる場合には、REPRO, CHANGE 制御文のNEWパラメータで指示を与える。即ち、入力側が区分編成で出力側が順編成であればNEW=PSを指示し、その逆の場合にはNEW=POを指示すれば良い。

・区分編成データセットのメンバの内容を、順編成データセットを新規作成して複写する。

```

// EXEC UPDATE, OF=▼A01.PO▼,
// NF=▼A01.PS▼, ORG=PS
//INPUT DD *
./ REPRO NAME=PODATA, NEW=PS

```

6.4.19 COMPR文

本文は順編成データセットどうし、区分編成データセットどうしの内容の比較を指示する文である。

(1) 記述形式

表 6.20

(2) 機能上の特記事項

(2.1) F1, F2パラメータで指定するデータセットの編成は同じでなければならない。

(2.2) 順編成データセットの場合は、2つのデータセット内のレコード数が全て等しく、対応するレコードおよびキーが全て等しいときに2つのデータセットは等しいとみなされる。区分編成データセットの場合は、1つのデータセットの全メンバが、他のデータセットに全て含まれ、対応するメンバのレコード数が全て等しく、対応するレコードおよびキーが全て等しく、ノートリストが存在するならばメンバ内のノートリストの位置が同じ時に2つのデータセットは等しいとみなされる。

(2.3) F1またはF2パラメータでデータセット名とメンバ名を指定すると、区分データセットの指定メンバを順データセットとして取扱うことができる。

(2.4) ORG=OPTを指定し、補助制御文を入力することにより次のことが可能となる。

- 補助制御文のLIMITパラメータでは比較中断条件として不一致レコード連続個数を指定できる。
- FIELDパラメータは、レコード全体の比較を行わない場合に、比較すべきレコード長とその先頭位置を指定できる。
- NAMEパラメータでは比較すべきメンバ名を指定できる。

(2.5) ORG=OPTを指定した時、補助制御文のカード入力を指示するために、INPUTなるdd名のDD文を必ず用意しなければならない。

(3) 使用例

(3.1) 順編成データセットどうしの全体を比較する。

```
// EXEC COMPR, F1=▼A01.A▼,
//      F2=▼A01.B▼
```

(3.2) 区分データセットの特定メンバどうしを比較する。

```
// EXEC COMPR,
//      F1=▼A01.A(PGM1)▼,
//      F2=▼A01.B(PGM1)▼
```

(3.3) 区分編成データセットの特定メンバどうしを、各レコードの先頭から72バイト、比較中断条件を20個として比較する。

```
// EXEC COMPR, ORG=OPT,
//      F1=▼A01.CASE1▼,
//      F2=▼A01.CASE2▼
//INPUT DD *
      COMPARE TYPORG=PO,
              LIMIT=20,
              FIELD=(72,1)          (i)
      MEMBER NAME=(DATA1,DATA2)    (ii)
```

(i) ORG=OPTを指定した場合にはCOMPARE制御文を省略できない。TYPORGパラメータでは比較するデータセットの編成を指定する。LIMITパラメータでは比較中断条件を指定する。またFI-

ELDパラメータでは比較レコードの長さと先頭位置を指定する。

(ii) MEMBER制御文のNAMEパラメータで比較すべきメンバ名を指定する。メンバ名が同じならば後のメンバ名は省略でき、括弧も不要となる。例えば

```
NAME=(A1,A2)
NAME=(A1,A2),(B1,B2)
NAME=A1,(B1,B2),C
```

等の記述ができる。

6.4.20 CRTODK文

(1) 記述形式

表 6.21

(2) 機能上の特記事項

(2.1) 入力データはカードイメージの書式付データまたはソースプログラムに限る。

(2.2) 入力データの中に、1, 2桁目が//または/*のカードは含めてはならない。

(2.3) データをカード入力することを指示するためにINPUTなるdd名のDD文を必ず用意しなければならない。

(2.4) NUMパラメータ省略時には入力データイメージで出力される。また指定があると73～80桁のデータは無効となり指定された順序番号が付加されて出力される。

(2.5) 第6.4.18項の(2.6)参照のこと。

(2.6) DISP=MODを指定し順編成データセットに追加書きする場合にNUMパラメータの指定があると、カード入力したデータに対してのみ順序番号が付加される。このため、同じ順序番号のデータが複数できてしまうことがあるからNUMパラメータの指定には注意を要する。全体に順序番号を付加する場合にはUPDATE文を用いる。

(3) 使用例

(3.1) 入力データの内容を新規作成の順編成データセットに保存する。

```
// EXEC CRTODK, NF=▼A01.DATA▼,
//      SPACE=▼(10,5)▼
//INPUT DD *
```

入力データカード

(3.2) 入力データの内容を既存の区分編成データセットに順序番号を付加して保存する。

```
// EXEC CRTODK, DISP=OLD,
//   NF=▼A01.SUM(A)▼, NUM=1
//INPUT DD *

```

(3.3) 入力データの内容を新規作成の区分編成データセットの2つのメンバに順序番号を付けて保存する。

```
// EXEC CRTODK, NF=▼A01.FPRO▼,
//   SPACE=▼(10,5,5)▼, NUM=3
//INPUT DD *
./ ADD NAME=A, LIST=NO      (i)
./ NUMBER NEW1=20, INCR=20
                                (ii)

./ ADD NAME=B, LIST=NO
./ NUMBER NEW1=20, INCR=20

```

(i) NAMEパラメータでメンバ名の指定を行う。
またLISTパラメータ省略時には制御文および入力データカードの全リストが出力される。

(ii) NEW1パラメータで順序番号初期値, INCRパラメータで増分の指定を行う。

6.4.21 CRTOLP文

(1) 記述形式

表 6.22

(2) 機能上の特記事項

第 6.4.20 項の (2.1), (2.2), (2.3), (2.4) を参照のこと。

(3) 使用例

(3.1) 入力したカードの内容に順序番号をつけてリスト出力する。

```
// EXEC CRTOLP, NUM=1
//INPUT DD *

```

(3.2) 入力したカードの内容を4部リスト出力する。

```
// EXEC CRTOLP, COPY=4
//INPUT DD *

```

(3.3) 入力したカードの内容に指定した順序番号を付けてリスト出力する。

```
// EXEC CRTOLP, NUM=3
//INPUT DD *
./ ADD
./ NUMBER NEW1=20, INCR=20
                                (i)

```

(i) NEW1パラメータで順序番号初期値を, INCRパラメータで増分を指定する。

6.4.22 CRTOCP文

(1) 記述形式

表 6.23

(2) 機能上の特記事項

第 6.4.21 項の(2)を参照のこと。

(3) 使用例

(3.1) 入力カードのコピーを3部とる。

```
// EXEC CRTOCP, COPY=3
//INPUT DD *

```

(3.2) 入力カードに順序番号を付けてコピーする。

```
// EXEC CRTOCP, NUM=1
//INPUT DD *

```

(3.3) 入力カードに指定した順序番号をつけてコピーする。

```
// EXEC CRTOCP, NUM=3
PUNCH CDSEQ=5, CDINCR=5
                                (i)
//INPUT DD *

```

(i) CDSEQパラメータで順序番号初期値を, CD-INCRパラメータで増分を指定する。

6.4.23 DKTODK文

(1) 記述形式

表 6.24

(2) 機能上の特記事項

(2.1) 入力データセット, 出力データセット共に1つのデータセットしか指定できない。

(2.2) 本文では順序番号付けとリスト出力が指定できない。その必要があればUPDATE文で処理すること。

(2.3) メンバ指定制御文のカード入力を指示するためにINPUTなるdd名のDD文を用意しなければならない。

(2.4) 出力データセットが区分編成の場合, DISPパラメータの指定はNEW, OLD以外は意味をもたない。出力側のデータセット内に複写メンバ名と同一のメンバ名が存在した場合, 更新を許すか否かはRパラメータで指定する。また, DISP=NEW指定時にはRパラメータは意味をもたない。

(2.5) 出力データセットが順編成の場合, データセットの処置はDISPパラメータで指定しなければならない。すなわち, データセットを新規作成して書込む場合にはNEW, 既存データセットの先頭から書込む場合にはOLD, 既存データセットに追加書きする場合にはMODを, それぞれ指定する。

(3) 使用例

(3.1) 順編成データセットの内容を新規作成の順編成データセットに複写する。なお, SPACEパラメータの値は省略値を用いる。

```
// EXEC DKTODK, OF=▼A01.MAIN▼,
//      NF=▼A01.MAIN1▼, ORG=3
```

(3.2) 区分編成データセットのメンバの内容を, 既存の順編成データセットに上書きする。

```
// EXEC DKTODK, DISP=OLD,
//      OF=▼A01.A(NO1)▼,
//      NF=▼A01.B▼, ORG=3
```

(3.3) 区分編成データセットのA1, A2, A3という3つのメンバを, 既存の区分編成データセ

ットにメンバ名を変更せずに複写する。複写先に同一のメンバ名が存在するときは置換えるものとする。

```
// EXEC DKTODK, OF=▼A01.PGM▼,
//      NF=▼A01.PGM1▼, MEM=ON,
//      R=ON, DISP=OLD
//INPUT DD *
S M=A1, A2, A3(註)
```

(註) メンバ指定制御文の説明は(3.4)を参照のこと。

(3.4) 区分編成データセットの複数メンバの内容を, 新規作成の区分編成データセットに複写する。

```
// EXEC DKTODK, OF=▼A01.A▼,
//      NF=▼A01.B▼,
//      SPACE=▼(10,5,5)▼, MEM=ON
//INPUT DD *
S M=A1, A2, A3 (i)
または
S M=(A1, B1), (A2, B2), (A3, B3) (ii)
または
E M=C1, C2, C3 (iii)
```

(i) 入力データセットの3つのメンバの内容を, 同一メンバ名で複写する。

(ii) 入力データセットの3つのメンバの内容を, 指定メンバ名で複写する。

(iii) 入力データセットの中の指定メンバを除く全メンバの内容を, 同一メンバ名で複写する。

6.4.24 DKTOLP文

(1) 記述形式

表 6.25

(2) 機能上の特記事項

(2.1) 入力データはカードイメージの書式付データまたはソースプログラムに限る。

(2.2) OFパラメータはOF, OF1, ..., OF4の順序で最大5個指定可能である。複数個指定する場合には第5.3.12項の(2.3)の条件を満たしていなければならない。また, 複数データセットに対して, COPY, MODパラメータは共通とする。

(2.3) OFパラメータで複数データセットを指

定した場合、システムはこれらを1つのデータセットとして扱うので、データセットの区切りで改頁することができない。

(2.4) MOD = EBCDICの指定ができるのは、レコード長が80バイトのデータに限る。80バイト以上のデータをリスト出力する場合にはMOD = HEXAを指定しなければならない。

(3) 使用例

(3.1) 区分編成データセットの4つのメンバの内容をリスト出力する。

```
// EXEC DKTOCP,
//   OF=▼A01.XY(X1)▼,
//   OF1=▼A01.XY(X2)▼,
//   OF2=▼A01.PP(P1)▼,
//   OF3=▼A01.PP(P2)▼
```

(3.2) 順編成データセットの内容を16進表現でリスト出力する。

```
// EXEC DKTOCP, OF=▼A01.LIST▼,
//   MOD=HEXA
```

6.4.25 DKTOCP文

(1) 記述形式

表 6.26

(2) 機能上の特記事項

第 6.4.20 項の (2.4), 第 6.4.24 項の (2.1), (2.2) 項参照のこと。

(3) 使用例

(3.1) 区分編成データセットの4つのメンバの内容をカード出力する。

```
// EXEC DKTOCP,
//   OF=▼A01.A(A1)▼,
//   OF1=▼A01.A(A2)▼,
//   OF2=▼A01.A(A3)▼,
//   OF3=▼A01.B(B1)▼
```

(3.2) 順編成データセットの内容を3部カード出力する。

```
// EXEC DKTOCP, COPY=3,
//   OF=▼A01.APTRAN▼
```

(3.3) 順編成データセットの内容に順序番号をつけてカード出力する。

```
// EXEC DKTOCP, OF=▼A01.PGM▼,
//   NUM=2
```

6.4.26 DKLIST文

(1) 記述形式

表 6.27

(2) 機能上の特記事項

(2.1) 本文実行時には、データセット毎にデータセット名、編成、レコード形式、レコード長、ブロック長、確保トラック数、使用中トラック数、最新参照日の情報と、指定ユーザの総確保トラック数、総使用トラック数の情報を出力する。

(2.2) NAMEパラメータで通常ユーザ名を指定するが、ユーザ名を含む修飾子を指定すると、その修飾子のついているデータセットに対して(2.1)で述べた情報を出力することができる。ただし、この場合には引用符で囲んで指定する。たとえば、NAME=▼A01.DATA▼と指定すると、A01.DATA.という名前のついたデータセットのみが対象となる。グループ名等を指定した場合も同じ。

(3) 使用例

(3.1) ユーザ名 A01 の個人ファイル上の全データセットに関する情報を出力する。

```
// EXEC DKLIST, NAME=A01
```

(3.2) ユーザ名 A01 の個人ファイル上のデータセットのうち、A01.DATA という修飾子のついたデータセットに関する情報を出力する。

```
// EXEC DKLIST,
//   NAME=▼A01.DATA▼
```

6.4.27 DRTYLIST文

(1) 記述形式

表 6.28

(2) 機能上の特記事項

(2.1) 指定できるデータセットは区分編成に限る。

(2.2) 指定データセットに対して、全メンバ名を出力する。

(3) 使用例

指定データセットの全メンバ名を出力する。

```
// EXEC DRTYLIST,
// FILE=▼A01.ABC▼
```

6.4.28 M7TOF 4文

(1) 記述形式

表 6.29

(2) 機能上の特記事項

MONITORⅦ用ジョブストリームをカード入力することを指示するためのDD文は不必要である。

(3) 使用例

本カタプロ文は次のように使用する。

```
//A01PASS JOB SHRT
// EXEC M7TOF4
//
// MONITORⅦ用
//   ジョブストリーム
```

6.4.29 USMT文

本文は申請ジョブ以外使用できない。

(1) 記述形式

表 6.30

(2) 機能上の特記事項

(2.1) 第 6.4.8 項の (2.5), 第 6.4.10 項の (2.7), (2.8) を参照のこと。

(2.2) SL形式のMTの場合、本文を参照モードで使用するときにはDISP=SHRを指定すると良い。この指定があるとたまたま他ユーザが同一のデータセット名を使用しているジョブの実行が待たされることがない。

(2.3) USMT文では常に自動的にPASS指定がなされている。PASS指定のある場合には、データセットのPASSとMTボリュームのPASSの2通りの意味をもつ。データセットのPASSの場合、次にそのデータセットを使用するジョブステップのUSMT文でRNO、データセット名のみを指定するだけでPASSされているデータセットを使用できる。また、ボリュームのPASSの場合には、RNO、データセッ

ト名、データセット順序番号、ボリューム通番を指定すれば、磁気テープの取付け作業を省くことができる。

(2.4) USMT文は実行を伴うカタプロ文(GOオプション指定のFORTC文、GO文、LINKGO文)の後で定義しなければならない。

(3) 使用例

(3.1) GO文の実行時に使用するデータセット識別番号が1のデータセットをダミー指定とする。

```
// EXEC GO,EF=▼A01.A▼,
// EMEM=B
// EXPAND USMT,RNO=01,
// DUMMY=ON
```

(3.2) GO文の実行時にデータセット識別番号2で出力されるデータを磁気テープ上の3番目に新規作成データセットとして書込む。

```
// EXEC GO,EF=▼A01.RJOB▼,
// EMEM=PGM1
// EXPAND USMT,RNO=02,
// FILE=DATA,FSEQ=3,
// DISP=NEW,VOL=NL0392
```

(3.3) GO文の実行時に、NL形式の磁気テープの5番目のデータセットをデータセット識別番号10で読込む。なお、NL形式の場合にもボリューム通番は必須であるから使用する本数だけ任意の名前のボリューム通番を指定する。

```
// EXEC GO,EF=▼A01.RJOB▼,
// EMEM=PGM2
// EXPAND USMT,RNO=10,
// FSEQ=5,BSIZE=500,
// RSIZE=50,RECFM=FB,
// VOL=A
```

(3.4) GOオプション指定のFORTC文実行時に、データセット識別番号90で出力されるデータをSL形式の磁気テープの先頭から新規作成データセットとして書込む。さらに、このデータセットを

次の GO 文実行時にデータセット識別番号 91 で使用する。なお、PASS されたデータセットを使用する場合には同一データセット名を指定しなければならない。

```
// EXEC FORTC, PARAM=GO,
//   SF=▼A01.SORT(P1)▼
// EXPAND USMT, RNO=90,
//   FILE=DATA5, DISP= NEW,
//   VOL=▼NM0012, NM0013▼
// EXEC GO, EF=▼A01.MARGE▼,
//   EMEM=P2
// EXPAND USMT, RNO=91,
//   FILE=DATA5
```

6.4.30 MTTOMT 文

本文は申請ジョブ以外使用できない。

(1) 記述形式

表 6.31

(2) 機能上の特記事項

(2.1) 入力側の磁気テープは、SL, NL 形式共に可能であるが、出力側は SL 形式に限る。

(2.2) 入力可能なデータセットの内容は、データ、ソースプログラム、オブジェクトモジュールに限る。

(2.3) OFSEQ で指定したボリューム上の位置に、OF で指定したデータセットがないとエラーとなる。

(2.4) 入力側が SL 形式の場合には OF パラメータを省略できない。なお、NF パラメータを省略した場合には OF と同じデータセット名で複写される。

(2.5) 本文は単一のデータセットしか扱えない。

(2.6) ボリューム上の全データセットを複写する場合には、MTCOPY 文を使用する。

(2.7) NV で指定したボリュームの処置は PASS 指定になっているため、後続のジョブステップでそのボリュームを使用する場合にはボリュームのセットを省くことができる。

(3) 使用例

(3.1) SL 形式の磁気テープの 3 番目のデータセットを、同じ名前での別の SL 形式の磁気テープの

4 番目に新規作成データセットとして複写する。

```
// EXEC MTTOMT, OF=MTDATA,
//   OFSEQ=3, OV=NL0312,
//   NFSEQ=4, NV=NM0090
```

(3.2) NL 形式の磁気テープの先頭のデータセットを、SL 形式の磁気テープの先頭のデータセットとして複写する。

```
// EXEC MTTOMT, OV=A,
//   NF=APDATA, NV=NS9326,
//   RSIZE=80, BSIZE=800,
//   RECFM=FB
```

6.4.31 MTCOPY 文

本文は申請ジョブ以外使用できない。

(1) 記述形式

表 6.32

(2) 機能上の特記事項

(2.1) PARAM=▼*▼指定の場合には単ボリュームの指定しかできない。

(2.2) OV パラメータで多ボリュームの指定をする場合、ラベル形式は統一されていなければならない。

(2.3) 出力は単ボリュームの指定に限る。もし出力途中で EOVS 状態となると、処理はそこで終了となるが、出力済みのデータセットは有効である。

(2.4) NL 形式の複写の場合には、出力側も NL 形式となる。

(2.5) PARAM=A を指定した場合、出力側では既存データセットと複写するデータセットのラベル形式が同じでなければならない。また、この指定があると DEN パラメータは無効となる。すなわち、記録密度を途中から変えることはできない。

(2.6) NL 形式の複写の場合、ボリューム通番は任意で良いが、必ず指定しなければならない。

(2.7) 出力側の磁気テープは、SL 形式の複写の場合には初期化済でなければならない。

(3) 使用例

(3.1) ボリューム通番 NL0029 の内容を、ボリューム通番 NL0030 にデッドコピーする。

```
// EXEC MTCOPY,OV=NL0029,
//   NV=NL0030
```

(3.2) ボリューム通番 NS0987 の内容を、ボリューム通番 NL0091 の内容の後に複写する。

```
// EXEC MTCOPY,OV=NS0987,
//   NV=NL0091,PARAM=A
```

(3.3) ボリューム通番 NM0001, NM0002, NM0003 の内容を、ボリューム通番 NL0001 の先頭から書込む。(下図の上参照)。

```
// EXEC MTCOPY,OV=▼NM0001,
//   NM0002,NM0003▼,
//   NV=NL0001,PARAM=H
```

(3.4) ボリューム通番 NM0001, NM0002, NM0003 の内容を、ボリューム通番 NL0001 の内容の後に書込む。(下図の下参照)。

```
// EXEC MTCOPY,OV=▼NM0001,
//   NM0002,NM0003▼,
//   NV=NL0001,PARAM=A
```

変換して個人ファイルに新規作成の区分編成データセットとして保存する。

```
//A01PASS JOB SHRT
// EXEC APTRAN,
//   TF=▼A01.CP(PGM1)▼,
//   SPACE=▼(10,5,5)▼
  APプログラムカード
//FT01 DD *
  制御データカード
//
```

6.5.2 翻訳のみを行うジョブ

〔例2〕 ソースプログラムをカード入力し翻訳を行う。この例のように、オブジェクトモジュールの保存や、LIED文への引渡しがない場合は、必ず NOOBJECT オプションを指定する。この指定がないと、システムはまったく無駄にオブジェクトモジュールを作成してしまう。

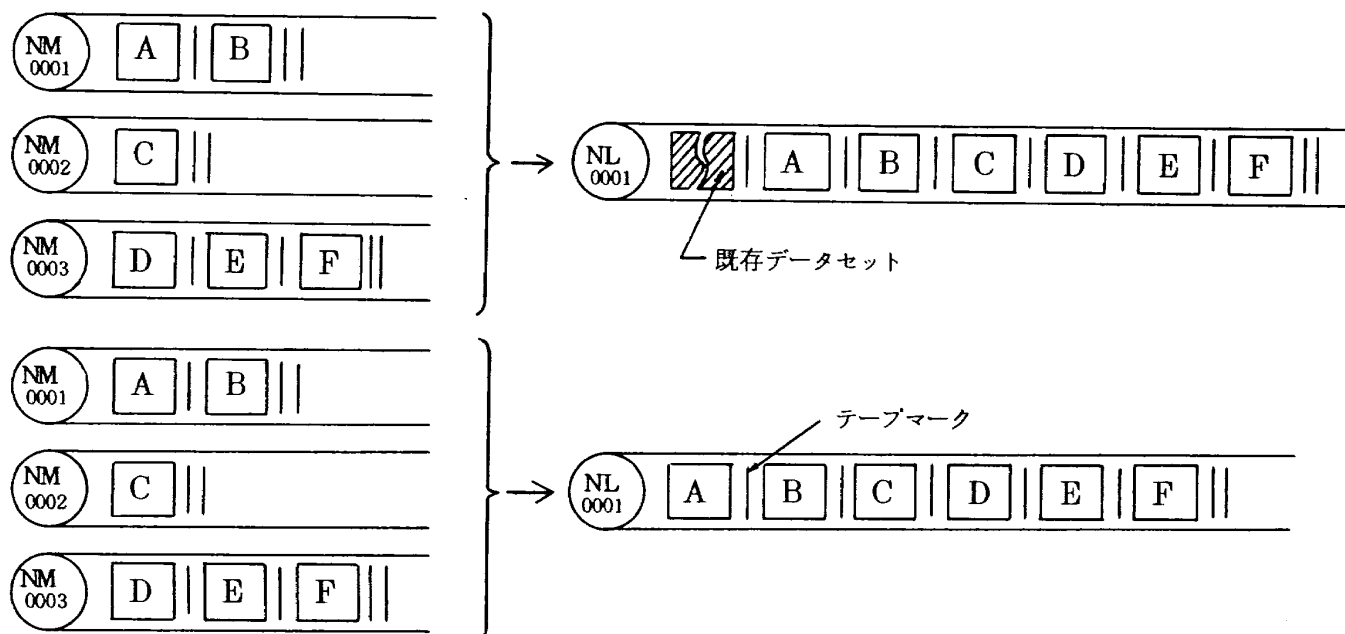
```
//A01PASS JOB SHRT
// EXEC FORTC,PARAM=NOOBJECT
  ソースプログラムカード
//
```

6.5 ジョブストリーム記述例

6.5.1 APプログラムの変換のみを行うジョブ

〔例1〕 APプログラムおよび変換時の制御データをカード入力し、FORTRAN 77 プログラムに

〔例3〕 個人ファイル上の区分データセットの全メンバとカード入力のソースプログラムを翻訳する。




```
//A01PASS JOB SHRT
// EXEC FORTC,PARAM=▼NOBJECT,
    ELM(*)▼,SF=▼A01.CP▼,
    SF1=▼A01.CP1▼
// EXEC FORTC,PARAM=NOBJECT
    ソースプログラムカード
//
```

6.5.3 APプログラムの変換・翻訳を行うジョブ

〔例4〕 APプログラムおよび変換時の制御データを個人ファイル及びカードから入力してFORT-RAN 77 プログラムに変換し、翻訳を行う。

```
//A01PASS JOB SHRT
// EXEC APTRAN,SF=▼A01.APU▼,
//    CF=▼A01.CONT▼
// EXEC APTRAN
    APプログラムカード
//FT01 DD *
    制御データカード
// EXEC FORTC,TS=IN,
//    PARAM=NOBJECT
//
```

6.5.4 ユーザライブラリを作成するジョブ

〔例5〕 ソースプログラムをカード入力して個人ファイル上の新規作成データセットにユーザライブラリとして登録する。また、個人ファイル上のソースプログラムも同一データセットに追加登録する。

```
//A01PASS JOB MIDL
// EXEC ULIB,EF=▼A01.LIB▼,
    SPACE=▼(10,10,5)▼
    ソースプログラムカード
// EXEC ULIB,EF=▼A01.LIB▼,
//    DISP=OLD,
//    SF=▼A01.A(SUB1)▼,
//    SF1=▼A01.A(SUB2)▼,
//    SF2=▼A01.B(SUB3)▼
//
```

6.5.5 翻訳・結合編集、実行を行うジョブ

〔例6〕 ソースプログラムをカード入力して、

翻訳から実行までを1ジョブステップで処理する。実行時使用データをカード入力する。

```
//A01PASS JOB MIDL
// EXEC FORTC,PARAM=GO
    ソースプログラムカード
//FT05 DD *
    入力データカード
//
```

〔例7〕 ソースプログラムを個人ファイルから入力して、翻訳・結合編集・実行を3ジョブステップで処理する。

```
//A01PASS JOB LONG
// EXEC FORTC,SF=▼A01.MAIN▼,
//    SF1=▼A01.SUB▼
// EXEC LIED
// EXEC GO
//
```

〔例8〕 ソースプログラムを個人ファイルおよびカードから入力し、翻訳・結合編集を行い、ロードモジュールを個人ファイルに保存するとともに、実行までの処理を行う。なお、実行時に参照する磁気テープデータをジョブ投入時にMT入力スプールする。

```
//A01PASS JOB MIDL
// EXEC FORTC,
//    SF=▼A01.MTJOB(PG3M)▼,
//    SF1=▼A01.PP(A)▼
// EXEC FORTC
    ソースプログラムカード
//L EXEC LIED,
//    OEF=▼A01.MTJOB(EB1)▼
// EXEC GO,SN=L
// EXPAND MTR,RNO=01,
//    FILE=DATA07,FSEQ=3,
//    VOL=NL0392
//
```

〔例9〕 既存のロードモジュールの一部を、カード入力したソースプログラムで修正し、同時に実

行まで行う。なお、実行時使用するデータを個人ファイルおよびカードから入力する。

```
//A01PASS JOB LONG
// EXEC FORTC
    ソースプログラムカード
//L1 EXEC LIED, IEF=▼A01.AB▼;
//      OEF=▼A01.AB(C)▼,
//      DISP=OLD
INCLUDE IEF(C)
// EXEC GO, SN=L1,
//      DF=▼A01.DATA▼
    入力データカード
//
```

〔例10〕 ソースプログラムを個人ファイルから入力し、翻訳・結合編集を行い、そのロードモジュールを異なるデータを用いて2度実行する。実行時使用データは個人ファイルから入力し、計算結果をMT出力スプール処理する。なお、磁気テープは同じものを使用する。

```
//A01PASS JOB LONG
// EXEC FORTC,
//      SF=▼A01.MTJOB(PG4M)▼,
//      SF1=▼A01.PP(A)▼
// EXEC LIED
// EXEC GO
// EXPAND USDK, RNO=01,
//      FILE=▼A01.DATA1▼
// EXPAND MTW, RNO=02,
//      FILE=DATA1, FSEQ=3,
//      VOL=NM0226
// EXEC GO
// EXPAND USDK, RNO=01,
//      FILE=▼A01.DATA2▼
// EXPAND MTW, RNO=02,
//      FILE=DATA2, FSEQ=4,
//      VOL=NM0226
//
```

〔例11〕 ソースプログラムをカード入力し、翻訳し、ローダで結合編集・実行を行う。なお、実

行時にワークファイルを使用する。

```
//A01PASS JOB SHRT
// EXEC FORTC
    ソースプログラムカード
// EXEC LINKGO,
//      ALIB=▼A01.LIB▼
// EXPAND USWK, RNO=10
//
```

〔例12〕 ソースプログラムをカード入力して翻訳・結合編集・実行までを1ジョブステップで実行する。実行時に参照・更新する磁気テープデータは入出力スプール処理する。また、実行時にXYプロッタ出力を行う。

```
//A01PASS JOB SHRT
// EXEC FORTC, PARAM=GO
    ソースプログラムカード
// EXPAND MTRW, RNO=02,
//      FILE=DATA3, DISP=OLD,
//      VOL=▼NM0001, NM0002▼
// EXPAND XY, SETNO=6
//
```

6.5.6 結合編集、実行を行うジョブ

〔例13〕 個人ファイル上のロードモジュールおよびユーザライブラリを入力し結合編集・実行を1ジョブステップで行う。

```
//A01PASS JOB MIDL
// EXEC LINKGO,
//      EF=▼A01.EBFILE(AB)▼,
//      ALIB=▼A01.LIB1▼,
//      ALIB1=▼A01.LIB2▼
//
```

〔例14〕 個人ファイル上のロードモジュールおよびユーザライブラリを入力して結合編集し、その出力を個人ファイルに保存するとともに、実行を行う。

```
//A01PASS JOB MIDL
//L EXEC LIED,S=ON,
//   IEF=▼A01.EBFILE▼,
//   ALIB=▼A01.LIB1▼,
//   ALIB1=▼A01.LIB2▼,
//   OEF=▼A01.EBFILE1(MEM1)▼,
//   SPACE=▼(5,2,3)▼
      INCLUDE IEF(AB)
// EXEC GO,SN=L
//
```

〔例 15〕 個人ファイル上のロードモジュールおよびユーザライブラリを入力して結合し、異なる実行時データを用いて 2 度実行する。なお、実行時データは、ジョブ投入時に MT 入力スプールする。

```
//A01PASS JOB LONG
// EXEC LIED,S=ON,
//   IEF=▼A01.MT1▼,
//   ALIB=▼A01.SUB▼
      INCLUDE IEF(MAIN)
// EXEC GO
// EXPAND MTR,RNO=01,FSEQ=2,
//   FILE=A,VOL=NM0001
// EXEC GO
// EXPAND MTR,RNO=01,FSEQ=3,
//   FILE=B,VOL=NM0001
//
```

6.5.7 実行を行うジョブ

〔例 16〕 個人ファイル上の実行形式プログラムを実行する。

```
//A01PASS JOB SHRT
// EXEC GO,EF=▼A01.ACC▼,
//   EMEM=XYPGM
//
```

〔例 17〕 複数の実行形式プログラムを実行する。なお実行時に作業用ファイルを使用する。

```
//A01PASS JOB SHRT
// EXEC GO,EF=▼A01.AB▼,
//   EMEM=PG1
// EXPAND USWK,RNO=01,
//   FILE=WORK1
// EXEC GO,EF=▼A01.AB▼,
//   EMEM=PG2
// EXPAND USWK,RNO=02,
//   FILE=WORK1,DISP=OLD
//
```

〔例 18〕 個人ファイル上の実行形式プログラムを実行し、ユーザジョブ処理自動継続機能を利用してチェックポイントをとってからジョブを終了する。

```
//A01PASS JOB LONG
//GO EXEC GO,EF=▼A01.P▼,
//   EMEM=P1,RD=NR
//CHECK DD DISP=NEW,
//   DSN=A01.CHECK,
//   SPACE=(TRK,(5,2),RLSE)
//
```

6.5.8 媒体変換を行うジョブ^註

〔例 19〕 入力データの内容を、新規作成の区分データセットの 2 つのメンバ A, B として保存する。

```
//A01PASS JOB SHRT
// EXEC CRTODK,
//   NF=▼A01.XYDATA▼,
//   SPACE=▼(10,3,2)▼, NUM=3
//INPUT DD *
./ ADD NAME=A
./ NUMBER NEW1=5,INCR=5
```

〔註〕 文献(9)で提示した通り、次期システムでは自動的にユーザプログラムの実行を伴うジョブと伴わないジョブ（即ち、処理プログラムまたはユーティリティのみを使用するジョブ）とを判別し、後者が常に迅速に処理されるように中央処理装置（CPU）を運用する。従って、媒体変換のようなユーティリティを使用するジョブはそれ自体で独立したジョブとして投入することにより、ジョブ処理のターンアラウンド時間が一段と短縮される。できるだけ、媒体変換文と実行文を一つのジョブストリームの中で一緒に記述すべきでなく、2 つのジョブに分離した方が非常に有利である。

```


./ ADD NAME=B
./ NUMBER NEW1=10, INCR=10

//

```

〔例20〕 区分編成データセットの指定メンバの内容を、既存の区分編成データセットに複写する。

```

//A01PASS JOB SHRT
// EXEC DKTODK, NF=▼A01.COPY▼,
//   OF=▼A01.MASTER▼,
//   R=ON, MEM=ON, DISP=OLD
// INPUT DD *
//   S M=(M1,C1),(M2,C2),
//       (M3,C3),(M4,C4)
//

```

〔例21〕 順編成データセットの内容を新規作成の順編成データセットに複写し、正しく複写されたことをCOMPR文で確認する。

```

//A01PASS JOB SHRT
// EXEC DKTODK, ORG=3,
//   OF=▼A01.PSDATA▼,
//   NF=▼A01.PSCOPY▼
// EXEC COMPR, F1=▼A01.PSDATA▼,
//   F2=▼A01.PSCOPY▼, ORG=PS
//

```

〔例22〕 複数の順編成データセットの内容を3部ずつリスト出力する。

```

//A01PASS JOB SHRT
// EXEC DKTOLP, OF=▼A01.PG1▼,
//   OF1=▼A01.PG2▼,
//   OF2=▼A01.PG3▼, COPY=3
//

```

〔例23〕 NL形式の磁気テープのデッドコピーを行う。

```

//A01PASS JOB SHRT
// EXEC MTCOPY, OV=A, NV=B
//

```

6.5.9 記述上の注意

APTRAN文、FORTC文、LIED文、GO文を複数枚含むジョブストリームを記述する際、各カタプロ間の間連性について次の注意を要する。

〔例24〕 ジョブストリームの記述例

```

//A01PASS JOB
// EXEC APTRAN
// EXEC APTRAN
// EXEC FORTC, TS=IN ①
// EXEC APTRAN
// EXEC FORTC, TS=IN
// EXEC LIED
// EXEC GO
// EXEC FORTC
//L EXEC LIED,
//   OEF=▼A01.A(B)▼
// EXEC GO, SN=L ②

```

〔例25〕 F4システムでは、APTRAN文の出力は、TS=IN指定のある最初のFORTC文にしが入力できない。また出力保存指定のあるLIED文の出力を後続のGO文で使用する場合には、LIED文のステップ名をGO文で指定しなければならない。そのため、上記の例の①、②の下線部のパラメータの指定を省略すると、以下のようなカタプロ間での入出力の受け渡しとなるので注意しなければならない。

```

//A01PASS JOB
// EXEC APTRAN
// EXEC APTRAN
// EXEC FORTC
// EXEC APTRAN
// EXEC FORTC, TS=IN
// EXEC LIED
// EXEC GO
// EXEC FORTC
//L EXEC LIED,
//   OEF=▼A01.A(B)▼
// EXEC GO

```

7. おわりに

現状では、計算機システムの移行に際し、ジョブの投入手続きを規定するジョブ制御言語は移行せざるを得ない。このことはユーザに対して、今まで慣れ親しんできたジョブストリーム記述法にかえて、新たなジョブストリームの記述法の習得を強いることを意味する。

本稿では、この移行に伴うユーザの負担を最小限にとどめ、かつ現用ジョブ制御マクロの改善を計るとともに、筆者らが提案した計算機の新規運用機能をも盛り込んだ次期計算機システム用ジョブ制御マクロの設計例を呈示した。

ここで提出したジョブ制御マクロ体系とそれらの設計仕様とは、どのようなオペレーティングシステムに対しても有効なものであるが、各ジョブ制御マクロの設計機能はオペレーティングシステムに依存して、本稿で記述した内容と若干異なることがある。

おわりに当り、ジョブ制御マクロ設計の試みに必要な資料の提出を頂いたファコム・ハイタック(株)並びに富士通(株)に対し、特に多くの討論を頂いた猪狩研一氏、畠間晴夫氏、森重博司氏、棚倉由行氏、磯辺文雄氏に対して、末筆ながら感謝の意を表す。

引用文献

- (1) 小野：今世紀最大の欠陥商品，bit，vol.12，No.13 (1980).
- (2) FACOM230 M-VL/VII “ジョブ制御言語文法書”
- (3) FACOM OS IV/X8 “ジョブ制御言語文法書”
- (4) FACOM OS IV/F4 “ジョブ制御言語文法書”
- (5) FACOM OS IV/X8 “FORTRAN 77 使用手引書”
- (6) FACOM OS IV/F4 “FORTRAN 77 使用手引書”
- (7) FACOM OS IV/X8 “リンケージエディタ／ローダ使用手引書”
- (8) FACOM OS IV/F4 “リンケージエディタ／ローダ使用手引書”
- (9) 畑山ら：次期航技研計算機システムの運用計画”，TM-430 (1981)
- (10) FACOM OS IV “HICS コミュニケータ使用手引書”
- (11) FACOM OS IV/F4 “RACF 使用手引書”
- (12) FACOM OS IV/F4 “JES/E 運用手引書”
- (13) FACOM OS IV/F4 “操作手引書”

表 6.2 JOB 文の記述形式

// ユーザ登録名 JOB [([ジョブクラス] [, [LP 出力打切行数] [, カード出力打切枚数])] [, TIME = { $\frac{\text{分}}{(\text{分}, \text{秒})}$ }] [, MSGCLASS = 0] [, COND = ((コード, 演算記号), ...)] [, TYPRUN = SCAN] [, NOTIFY = ユーザ名] [, SPARAM = ▼ YY.MM.DD ▼] [, RESTART = (GO 文のステップ名, GO, POINT)] [, GROUP = グループ名]		
パラメータ	省略値	パラメータの説明
ジョブクラス	SHRT	ジョブクラス (SHRT, MIDL, LONG, LARG) を指定する。
LP 出力 打切行数	センタ設定値	LP 出力打切論理行数を指定する。
カード出力 打切枚数	〃	カード出力打切枚数を指定する。
TIME	〃	ジョブ打切時間を指定する。ただし、分は 0 ～ 1440、秒は 0 ～ 59 の範囲 内で、センタ設定値以下でなければならない。
MSGCLASS	システム標準値	ジョブの出力先をオープン出力とする場合に指定する。
COND	〃	コードと演算記号の対で、ジョブ終了条件を指定する。条件は 8 個まで指 定できる。指定コードは完了コードと比較され、1 つでも条件を満たすと その時点でジョブ処理は終了する。
TYPRUN	——	ジョブ制御文のシンタックスチェックのみを行う場合に指定する。
NOTIFY	——	ジョブの実行終了メッセージを TSS 端末に出力させる場合に、TSS ユー ザ名を指定する。
SPARAM	▼ 実行日 ▼	ジョブ実行中に TIME マクロ命令を発行した時にプログラムに返される実 際の実行日を変更したい場合に指定する。YY に西暦年度の下 2 桁、MM に月、DD に日を指定する。
RESTART	——	ユーザジョブ処理自動継続機能を利用してジョブ処理を再開する場合に指 定する。ただし LONG と LARG 以外のジョブクラスでは利用できない。
GROUP	デフォルト グループ名	カレントグループのグループ名を指定する (文献[1]参照)。

表 6.3 APTRAN 文の記述形式

<pre>//[ステップ名] EXEC APTRAN [,PARAM=▼パラメータ▼][,SF=▼データセット名[(メンバ名)]▼] [,CF=▼データセット名[(メンバ名)]▼] [,TF=▼データセット名[(メンバ名)]▼] [,DISP={$\begin{matrix} \text{NEW} \\ \text{OLD} \\ \text{MOD} \end{matrix}$}^(注)][,ORG={$\begin{matrix} \text{PO} \\ \text{PS} \end{matrix}$}] [,SPACE=▼([トラック数][, [増分][,ディレクトリブロック数]) [,RLSE]▼]</pre>		
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	APTRAN のオプションを指定する。
SF	——	AP プログラムを個人ファイルから入力する場合に指定する。区分編成の場合メンバ名は省略できない。
CF	——	APTRAN の制御データを個人ファイルから入力する場合に指定する。区分編成の場合メンバ名は省略できない。
TF	——	APTRAN の出力を個人ファイルに保存する場合に指定する。区分編成の場合メンバ名は省略できない。
DISP	NEW	TF で指定したデータセットの処置を指定する。表 6.1 参照
ORG	PO	TF で指定したデータセットを DISP = NEW で新規作成する場合、このパラメータで編成を指定すれば SPACE パラメータの指定を省略できる。
SPACE	センタ設定値	TF で指定したデータセットの領域確保に関するサブパラメータを指定する。第 6.3.5 項参照。

(注) 細下線は指定を省略したときに、システムがセットする値を示す(以下同様)。

表 6.4 FORTC 文の記述形式

<pre> //[ステップ名] EXEC FORTC [,PARAM=▼パラメータ▼][,TS= IN] [,SF=▼データセット名[(メンバ名)]▼] [,RF=▼データセット名[(メンバ名)]▼]^(注1) [,DISP={$\begin{matrix} \text{MEW} \\ \text{OLD} \\ \text{MOD} \end{matrix}$}][,ORG={$\begin{matrix} \text{PO} \\ \text{PS} \end{matrix}$}] [,SPACE=▼([トラック数][, [増分][,ディレクトリブロック数]]) [,RLSE]▼] [,COND=([(コード, 演算記号 [, ステップ名]), ...] [,EVEN])] [,SINC=▼データセット名▼][,PSP=ON]^(注2) </pre>		
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	FORTTRAN 77 コンパイラオプションを指定する。
TS	——	出力保存指定のない先行のAPTRAN文の出力を全て連結して当カタプロ文の入力とする場合に指定する。
SF	——	個人ファイル上のソースプログラムを入力する場合に指定する。複数個指定可能である。 ^(注3)
RF	——	FORTC文の出力を個人ファイル上に保存する場合に指定する。区分編成とする場合にはメンバ名は省略できない。
COND	——	<p>当カタプロ文を実行しない条件を指定する。指定コードは指定ステップの完了コードと比較される。ステップ名を省略した場合には、先行の全カタプロ文が比較の対象となる。</p> <p>EVEN：先行カタプロ文の終了状態によらず当カタプロ文を実行する指示。</p> <p>ONLY：先行カタプロ文のどれかが異常終了した時のみ当カタプロ文を実行する指示。</p> <p>演算記号；第 6.4.1 項の (2.6) 参照</p>
SINC	——	*INCLUDE行で記述したメンバが含まれるデータセットを指定する。 ^(注3) 複数個指定可能である。
PSP	——	X8システムのXYを使用する場合に指定する。本パラメータはGOオプションの指定がある場合にのみ意味をもつ。

(注1) 太下線のパラメータはできるだけ使用しないこと。

(注2) X8システムのXYを使用するユーザ以外使用してはならない。

(注3) SFパラメータは、SF, SF1, ..., SF4の5個、SINCパラメータはSINC, SINC1, ..., SINC4の5個がそれぞれ使用できる。

(注4) DISP, ORG, SPACEパラメータは、RFパラメータで指定したデータセットの処置、編成、領域に関するパラメータであり、APTRAN文に準ずる。

表 6.5 LIED 文の記述形式

<pre> // [ステップ名] EXEC LIED [, PARAM=▼パラメータ▼] [, S=ON] [, RF=▼データセット名 [(メンバ名)] ▼]^(注1) [, ALIB=▼データセット名▼] [, IEF=▼データセット名▼] [, OEF=▼データセット名 [(メンバ名)] ▼] [, DISP= { NEW / OLD / MOD }] [, SPACE=▼ ([トラック数] [, [増分] [, ディレクトリブロック数])] [, RLSE] ▼] [, R=ON] [, COND= ([(コード, 演算記号 [, ステップ名]) , ...] [, EVEN])]^(注2) [, PSP=ON] </pre>		
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	リンケージエディタオプションを指定する。
S	——	先行 FORTC 文からのオブジェクトモジュールを受取らないか、又は受取るものがない場合に指定する。
RF	——	個人ファイル上のオブジェクトモジュールを入力する場合に指定する。区分編成の場合メンバ名は省略できない。複数個指定可能である。 ^(注3)
ALIB	——	個人ファイル上のロードモジュールを自動呼出ライブラリとする場合に指定する。複数個指定可能である。 ^(注3)
IEF	——	個人ファイル上のロードモジュールを入力する場合に指定する。メンバ名は INCLUDE 文で指定しなければならない (使用例参照)。複数個指定可能である。 ^(注3)
OEF	——	LIED 文の出力を個人ファイル上に保存する場合に指定する。
R	——	ジョブストリームの中に OEF パラメータの指定がない LIED 文が複数ある場合、2 枚目以降の LIED 文に R=ON を指定しなければならない。なお、OEF パラメータの指定がない最初の LIED 文には本パラメータを指定してはならない。
PSP	——	X8 システムの XY を使用する場合に指定する。

(注1) 太下線のパラメータはできるだけ使用しないこと (第 3.2.1 項参照)。

(注2) X8 システムの XY を使用するユーザ以外使用してはならない。

(注3) RF パラメータは RF, RF1, …, RF4 の 5 個, ALIB パラメータは ALIB, ALIB1, …, ALIB4 の 5 個, IEF パラメータは IEF, IEF1, …, IEF 4 の 5 個がそれぞれ使用できる。

(注4) DISP, SPACE パラメータは OEF パラメータで指定したデータセットの処置, 領域に関するパラメータであり, APTRAN 文に準ずる。また, COND パラメータは FORTC 文に同じである。

表 6.6 GO文の記述形式

<pre> [/[ステップ名] EXEC GO [,PARAM=▼パラメータ▼][,EF=▼データセット名▼ ,EMEM=メンバ名] [,DF=▼データセット名[(メンバ名)]▼][,TIME={ (分,秒) }] [,COND=([(コード, 演算記号[,ステップ名]), ...] [,EVEN])] [,RD=NR][,SN=ステップ名] </pre>		
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	実行可能プログラムオプションを指定する。
EF	——	実行プログラムを個人ファイルから入力する場合に指定する。
EMEM	——	EFで指定したデータセットに含まれる実行プログラムのメンバ名を指定する。
DF	——	実行時使用データを個人ファイルから入力する場合に指定する。複数個指定可能である。 ^(注1)
TIME	センタ設定値	当カタプロ文の処理打切時間を指定する。
RD	——	ユーザジョブ処理自動継続機能を利用してチェックポイントをとる場合に指定する。ただし、本パラメータはLONGとLARGジョブクラスでのみ指定できる。
SN	——	出力保存指定のあるLIED文の出力を、自動的に本カタプロ文の入力としたい場合に、そのLIED文のステップ名を指定する。

(注1) DFパラメータは、DF, DF1, ..., DF4の5個使用できる。

(注2) CONDパラメータはFORTC文に同じである。

表 6.7 LINKGO 文の記述形式

<pre> //[ステップ名] EXEC LINKGO [,PARAM=▼パラメータ▼] (注1) [, { $\frac{RF}{EF}$ } = ▼データセット名 [(メンバ名)] ▼] [,ALIB=▼データセット名▼][,DF=▼データセット名 [(メンバ名)] ▼][,TIME= { $\frac{分}{(分,秒)}$ }] [,COND=([(コード, 演算記号 [,ステップ名]), ...] [,EVEN])] [,PSP=ON] (注2) </pre>		
パラメータ	省略値	パラメータの説明
PARAM	システム標準値	ローダオプションを指定する。
RF	——	個人ファイル上のオブジェクトモジュールを主要入力とする場合に指定する。区分編成の場合メンバ名は省略できない。複数個指定可能である。(注3)
EF	——	個人ファイル上のロードモジュールを主要入力とする場合に指定する。メンバ名は省略できない。複数個指定可能である。(注3)

(注1) 太下線のパラメータはできるだけ使用しないこと(第3.2.1項参照)。

(注2) X8システムのXYを使用するユーザ以外は使用してはならない。

(注3) RFパラメータは, RF, RF1, ..., RF4の5個, EFパラメータはEF, EF1, ..., EF4の5個, ALIBパラメータはALIB, ALIB1, ..., ALIB4の5個, DFパラメータはDF, DF1, ..., DF4の5個がそれぞれ使用できる。

(注4) CONDパラメータはFORTC文に, ALIB, PSPパラメータはLIED文に, DF, TIMEパラメータはGO文に同じである。

表 6.8 ULIB 文の記述形式

<pre> //[ステップ名] EXEC ULIB[,FPARAM=▼パラメータ▼][,SF=▼データセット名 [(メンバ名)] ▼] [,SINC=▼データセット名▼][,LPARAM=▼パラメータ▼] ,EF=▼データセット名▼[,DISP= { $\frac{NEW}{OLD}$ }] [,SPACE=▼([トラック数] [,増分] [,ディレクトリブロック数])] [,RLSE] ▼] </pre>		
パラメータ	省略値	パラメータの説明
FPARAM	システム標準値	FORTTRAN 77 コンパイラオプションを指定する。
LPARAM	〃	リンケージエディタオプションを指定する。
EF	——	ユーザライブラリの保存先データセット名を必ず指定する。

(注1) SFパラメータは, SF, SF1, ..., SF4の5個, SINCパラメータはSINC, SINC1, ..., SINC4の5個が指定できる。

(注2) DISP, SPACEパラメータはEFパラメータで指定したデータセットの処置, 領域に関するパラメータでありAPTRAN文に準ずる。また, SF, SINCパラメータはFORTC文に同じである。

表 6.9 USDK 文の記述形式

<pre>// EXPAND USDK, RNO=データセット識別番号[, DUMMY=ON] [, FILE=▼データセット名(メンバ名)▼][, DISP={ NEW OLD MOD }] [, ORG={ PO PS }][, SPACE=▼(トラック数)[, [増分] [, ディレクトリブロック数]][, RLSE]▼][, ACC= READ] [, BSIZE=ブロック長][, RSIZE=レコード長][, RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータの説明
RNO	——	プログラム内入出力文のデータセット識別番号を指定する。
DUMMY	——	実際の入出力動作が不要である場合に指定する。
FILE	——	使用するデータセット名を指定する。区分編成の場合メンバ名は省略できない。複数個指定可能である。 ^(注)
DISP	OLD	使用するデータセットの処置を指定する。表 6.1 参照。
ORG	PO	FILEパラメータで指定したデータセットをDISP=NEWで新規作成する場合、このパラメータで編成を指定すればSPACEパラメータの指定を省略できる。
SPACE	センタ設定値	使用するデータセットの領域に関するサブパラメータを指定する。第 6.3.5 項参照。
ACC	——	使用するデータセットが書き込み保護されている場合に指定する。
BSIZE	——	DISP=NEWでデータセットを新規作成する場合にブロック長を指定する(省略可能である)。
RSIZE	——	DISP=NEWでデータセットを新規作成する場合にレコード長を指定する(省略可能である)。
RECFM	——	DISP=NEWでデータセットを新規作成する場合にレコード形式を指定する(省略可能である)。

(注) FILEパラメータは、FILE, FILE1, ..., FILE4 の 5 個指定できる。

表 6.10 USWK 文の記述形式

<pre>// EXPAND USWK, RNO=データセット識別番号[, DUMMY=ON][, FILE=データセット名] [, SPACE=▼トラック数, 増分▼][, DISP={ NEW OLD }] [, BSIZE=ブロック長][, RSIZE=レコード長] [, RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータの説明
FILE	WORKFILE	ワークファイルのデータセット名を指定する。

(注) 上記以外のパラメータの意味は、USDK 文に同じである。

表 6.11 CF 文の記述形式

<pre>// EXPAND CF[,UNIT={^(注)<u>MT</u> <u>DK</u>}][,FILE=▼データセット名[(メンバ名)]▼] [,FSEQ=データセット順序番号][,VOL=▼ボリューム通番,...▼] [,BSIZE=ブロック長][,RSIZE=レコード長] [,RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータの説明
UNIT	DK	使用装置名を指定する。
FILE	——	個人ファイルを使用する場合、または SL 形式の磁気テープを使用する場合に指定する。NL 形式の磁気テープを使用する場合には指定してはならない。
FSEQ	1	磁気テープを使用する場合に、使用するデータセットのボリューム上の位置を指定する。
VOL	——	磁気テープを使用する場合に、使用する本数のボリューム通番を指定する。
BSIZE	——	NL 形式の磁気テープを使用する場合に必ず指定する。
RSIZE	——	〃
RECFM	——	〃

(注) 太下線のパラメータは申請ジョブ以外使用できない。

表 6.12 MTR 文の記述形式

<pre>// EXPAND MTR,RNO=データセット識別番号[,FILE=▼データセット名▼] [,FSEQ=データセット順序番号],VOL=▼ボリューム通番,...▼ [,BSIZE=ブロック長][,RSIZE=レコード長] [,RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータの説明
RNO	——	ユーザプログラム内の入力文のデータセット識別番号を指定する。
FILE	——	磁気テープが SL 形式の場合に指定する。NL 形式の場合指定してはならない。

(注) 上記以外のパラメータは、CF 文に同じである。

表 6.13 MTW文の記述形式

<pre>// EXPAND MTW, RNO=データセット識別番号, FILE=▼データセット名▼ [, FSEQ=データセット順序番号], VOL=▼ボリューム通番, ...▼ [, BSIZE=ブロック長][, RSIZE=レコード長] [, RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータ説明
RNO	——	ユーザプログラム内の出力文のデータセット識別番号を指定する。
FILE	——	使用する磁気テープのデータセット名を指定する。
BSIZE	——	書込時のブロック長を指定する。
RSIZE	——	書込時のレコード長を指定する。
RECFM	——	書込時のレコード形式を指定する。

(注) 上記以外のパラメータはCF文に同じである。

表 6.14 MTRW文の記述形式

<pre>// EXPAND MTRW, RNO=データセット識別番号[, FILE=▼データセット名▼] [, FSEQ=データセット順序番号], VOL=▼ボリューム通番, ...▼ [, DISP={<u>OLD</u> MOD}][, BSIZE=ブロック長] [, RSIZE=レコード長][, RECFM=レコード形式]</pre>		
パラメータ	省略値	パラメータの説明
RNO	——	ユーザプログラム内の入出力文のデータセット識別番号を指定する。
DISP	MOD	使用データセットの実行時の最初のアクセス位置を指定する。 OLD: データセットの先頭を指示する場合。 MOD: データセットの最後を指示する場合。

(注) FILEパラメータはMTR文に、その他のパラメータはCF文に同じである。

表 6.15 CP文の記述形式

// EXPAND CP[, DUMMY=ON]		
パラメータ	省略値	パラメータの説明
DUMMY	——	実際にカード出力を行わない場合に指定する。

表 6.16 PP文の記述形式

// EXPAND PP[, DEBUG=ON]		
パラメータ	省略値	パラメータの説明
DEBUG	——	実際の出力は行わない場合に指定する。

表 6.17 XY文の記述形式

// EXPAND XY[,DEBUG=ON][,SETNO=番号][,XYP=ON][XYF=▼データセット名▼] [,DISP= { $\frac{\text{NEW}}{\text{OLD}}$ }] ^(注1)		
パラメータ	省略値	パラメータの説明
DEBUG	——	実際の出力は行わない場合に指定する。
SETNO	1	オペレータがセットすべき用紙やペンの種類を指定する。 ^(注2)
XYP	——	ジョブ終了後、プロッタ出力を自動的にX8システムに伝送する場合に指定する。
XYF	——	X8システムのXYプロッタ出力を一旦個人ファイルに保存する場合にデータセット名を指定する。ただし順編成に限る。
DISP	NEW	XYFで指定したデータセットの処置を指定する。

(注1) 太下線のパラメータは、X8システムのXYを使用するユーザ以外指定してはならない。

(注2) SETNOは1～8まで指定でき、下表の意味をもつ。

種類 \ SETNO	1	2	3	4	5	6	7	8
紙幅 { NARR: 25cm WIDE: 83cm }	NARR	NARR	NARR	NARR	WIDE	WIDE	WIDE	WIDE
用紙 { 00 : 白紙 02 : 目盛 }	00	00	02	02	00	00	02	02
ペン { BALL: ボールペン INK : インク }	BALL	INK	BALL	INK	BALL	INK	BALL	INK

表 6.18 PPXY文の記述形式

```
// EXPAND PPXY[ ,DEBUG=ON][ ,SETNO=番号 ]
```

(注) パラメータの意味はXY文に同じである。

表 6.19 UPDATE文の記述形式

<pre>//[ステップ名] EXEC UPDATE[,OF=▼データセット名[(メンバ名)]▼] [,NF=▼データセット名[(メンバ名)]▼][,DISP={NEW OLD MOD}] [,ORG={PO PS}][,SPACE=▼([トラック数][,[増分] [,ディレクトリブロック数]])[,RLSE]▼]</pre>		
パラメータ	省略値	パラメータの説明
OF	——	更新するデータセット名を指定する。区分編成の場合、メンバ名は当パラメータか機能制御文で必ず指定する。
NF	——	更新データセットの保存先データセット名を指定する。OFで指定したデータセット名と同じ場合には指定してはならない。
DISP	NEW	NFで指定したデータセットの処置を指定する（表 6.1 参照）。
ORG	PO	次の条件を満たす場合には、NFで指定したデータセットの編成を指定しなければならない。 <ul style="list-style-type: none"> ・順編成データセットを更新し、もとのデータセットに上書きする場合。 ・SPACEパラメータを省略してDISP=NEWでデータセットを新規作成する場合。
SPACE	センタ設定値	NFで指定したデータセットの領域に関するサブパラメータを指定する（第 6.3.5 項参照）。

表 6.20 COMPR文の記述形式

<pre>//[ステップ名] EXEC COMPR,F1=▼データセット名[(メンバ名)]▼ ,F2=▼データセット名[(メンバ名)]▼[,ORG={PO PS OPT}]</pre>		
パラメータ	省略値	パラメータの説明
F1, F2	——	内容の比較を行うデータセット名を指定する。当パラメータでメンバ名までを指定すると、順編成として扱うことができる。
ORG	PS	データセットの編成および補助制御文入力の有無を指定する。 PS : 順編成データセット全体の比較を行うが、補助制御文は入力しない場合の指定。 PO : 区分編成データセット全体の比較を行うが、補助制御文は入力しない場合の指定。 OPT : 補助制御文の入力を行う場合の指定。

表 6.21 CRTODK 文の記述形式

$//[\text{ステップ名}] \text{ EXEC CRTODK, NF} = \nabla \text{データセット名} [(\text{メンバ名})] \nabla [, \text{DISP} = \left\{ \begin{array}{c} \text{NEW} \\ \text{OLD} \\ \text{MOD} \end{array} \right\}]$ $[, \text{ORG} = \left\{ \begin{array}{c} \text{PO} \\ \text{PS} \end{array} \right\}] [, \text{SPACE} = \nabla ([\text{トラック数}] [, [\text{増分}]$ $[, \text{ディレクトリブロック数}])] [, \text{RLSE}] \nabla]$ $[, \text{NUM} = \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right\}] [, \text{LIST} = \text{YES}]$		
パラメータ	省略値	パラメータの説明
NF	——	保存先データセット名を指定する。区分編成の場合、メンバ名は当パラメータか補助制御文で必ず指定する。
NUM	——	個人ファイルに保存する際、入力データに順序番号を付加する場合に指定する。 1：初期値 = 10，増分値 = 10 の指定。 2：初期値 = 100，増分値 = 100 の指定。 3：順序番号付けの指定は補助制御文で行う指定。
LIST	——	リスト出力を必要とする場合に指定する。

(注) DISP, ORG, SPACE パラメータは NF で指定したデータセットの処置、編成、領域に関するパラメータであり、USDK 文に準ずる。

表 6.22 CRTOLP 文の記述形式

$//[\text{ステップ名}] \text{ EXEC CRTOLP} [, \text{COPY} = \text{コピー部数}] [, \text{NUM} = \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right\}]$		
パラメータ	省略値	パラメータの説明
COPY	1	コピー部数を指定する。
NUM	——	順序番号をつけて出力をする場合に指定する（表 6.21 参照）。

表 6.23 CRTOCP 文の記述形式

$//[\text{ステップ名}] \text{ EXEC CRTOCP} [, \text{COPY} = \text{コピー部数}] [, \text{NUM} = \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \right\}]$		
--	--	--

(注) パラメータの意味は CRTOLP 文に同じである。

表 6.24 DKTODK 文の記述形式

<pre>//[ステップ名] EXEC DKTODK,OF=▼データセット名[(メンバ名)]▼ ,NF=▼データセット名[(メンバ名)]▼[,DISP={NEW OLD MOD}] [,ORG={1 2 3}][,R=ON][,SPACE=▼([トラック数] [, [増分][,ディレクトリブロック数]))[,RLSE]▼][,MEM=ON]</pre>		
パラメータ	省略値	パラメータの説明
OF	——	入力データセット名を指定する。区分編成でMEM=ON指定のない場合にはメンバ名は省略できない。
NF	——	出力データセット名を指定する。区分編成でMEM=ON指定のない場合にはメンバ名は省略できない。
ORG	1	入力および出力データセットの編成を指定する。 1：入力が区分編成で、出力が区分編成の場合。 2：入力が順編成で、出力が区分編成の場合。 3：入力が区分編成または順編成で、出力が順編成の場合。
R	——	区分編成データセットへ複写する場合で、出力メンバがNFで指定したデータセット内に存在したら置き換える場合に指定する。
MEM	——	メンバ指定制御文を入力する場合に指定する（使用例参照）。

(注) DISP,SPACEパラメータは、NFで指定したデータセットの処置、領域に関するパラメータであり、APTRAN文に準ずる。

表 6.25 DKTOLP 文の記述形式

<pre>//[ステップ名] EXEC DKTOLP,OF=▼データセット名[(メンバ名)]▼[,COPY=コピー部数] [,MOD={EBCDIC HEXA}]</pre>		
パラメータ	省略値	パラメータの説明
OF	——	リスト出力するデータセット名を指定する。区分編成の場合にはメンバ名は省略できない。複数個指定可能である。 ^(注1)
MOD	EBCDIC	リストの出力形式を指定する。

(注1) OFパラメータはOF,OF1, …, OF4の5個が使用できる。

(注2) COPYパラメータはCRTOLP文に同じである。

表 6.26 DKTOCP 文の記述形式

//[ステップ名] EXEC DKTOCP, OF=▼データセット名[(メンバ名)]▼ [, COPY=コピー部数][, NUM= $\begin{Bmatrix} 1 \\ 2 \\ 3 \end{Bmatrix}$]		
---	--	--

(注1) OFパラメータはOF, OF1, ..., OF4の5個が使用できる。

(注2) OFパラメータはDKTOLP文, COPY, NUMパラメータはCRTOLP文に同じである。

表 6.27 DKLIST 文の記述形式

//[ステップ名] EXEC DKLIST, NAME=データセットの第1修飾子名		
パラメータ	省略値	パラメータの説明
NAME	——	データセットの第1修飾子名を指定する。なお、第1修飾子名を含む修飾子を記述する場合には引用符で囲む。

表 6.28 DRTYLIST 文の記述形式

//[ステップ名] EXEC DRTYLIST, FILE=▼データセット名▼		
パラメータ	省略値	パラメータの説明
FILE	——	区分編成のデータセット名を指定する。

表 6.29 M7TOF4 文の記述形式

// EXEC M7TOF4	パラメータなし
----------------	---------

表 6.30 USMT 文の記述形式

// EXPAND USMT, RNO=データセット識別番号[, DUMMY=ON][, FILE=▼データセット名▼] [, DISP= $\begin{Bmatrix} \text{NEW} \\ \text{OLD} \\ \text{MOD} \\ \text{SHR} \end{Bmatrix}$][, FSEQ=データセット順序番号] [, VOL=▼ボリューム通番, ...▼][, BSIZE=ブロック長] [RSIZE=レコード長][, RECFM=レコード形式]		
---	--	--

(注) RNO, DUMMY, DISPパラメータはUSDK文に、その他のパラメータはCF文に準ずる。

表 6.31 MTTOMT 文の記述形式

// [ステップ名] EXEC MTTOMT [, OF = ▼データセット名 ▼] , OFSEQ = データセット順序番号 , OV = ▼ボリューム通番, ... ▼ [, BSIZE = ブロック長] [, RSIZE = レコード長] [, RECFM = レコード形式] [, NF = ▼データセット名 ▼] , NFSEQ = データセット順序番号 , NV = ▼ボリューム通番, ... ▼ [, DISP = $\begin{Bmatrix} \text{NEW} \\ \text{OLD} \\ \text{MOD} \end{Bmatrix}$]		
パラメータ	省略値	パラメータの説明
OF	——	入力データセット名を指定する。NL形式の磁気テープの場合指定してはならない。
OFSEQ	1	入力データセットのボリューム上の位置を指定する。
OV	——	入力磁気テープのボリューム通番を、使用する本数だけ指定する。
NF	——	出力データセット名を指定する。OFパラメータ指定時に当パラメータを省略するとOFと同じとみなされる。
NFSEQ	1	出力データセットのボリューム上の位置を指定する。
NV	——	出力磁気テープのボリューム通番を使用する本数だけ指定する。

(注) DISPパラメータはUSDK文に、その他のパラメータはCF文に準ずる。

表 6.32 MTCOPY 文の記述形式

// [ステップ名] EXEC MTCOPY , OV = ▼ボリューム通番, ... ▼ , NV = ボリューム通番 [, PARAM = $\begin{Bmatrix} \text{▼*▼} \\ \text{H[EAD]} \\ \text{A[FTER]} \end{Bmatrix}$] [, DEN = $\begin{Bmatrix} 2 \\ 3 \\ 4 \end{Bmatrix}$]		
パラメータ	省略値	パラメータの説明
OV	——	入力側のボリューム通番を指定する。
NV	——	出力側のボリューム通番を指定する。複数ボリュームの指定は行えない。
PARAM	▼*▼	コピーの条件を指定する。 ▼*▼ : 入力ボリュームの全内容のデッドコピーの指示。この場合には単ボリュームの指定しかできない。 HEAD : OVパラメータで指定したボリューム内の全データセットを出力ボリュームの先頭から書込む指示。 AFTER : OVパラメータで指定したボリューム内の全データセットを出力ボリュームの既存データセットの後に書込む指示。
DEN	——	出力時の記録密度を2～4の数字で指定する。省略時には磁気テープ装置のもつ最高密度で出力される。 2 : 800BPI 3 : 1600BPI 4 : 6250BPI

航空宇宙技術研究所資料444号

昭和56年9月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182

印刷所 株式会社 東京プレス
東京都板橋区桜川2-27-12

Printed in Japan