

《解説》

OUK 9700 II EXEC 7 のジョブ制御について

中川 一郎* 中江 康史*

1. 概説

ジョブ制御プログラムは非常駐プログラムであり、システム資源（主記憶装置、ソフトウェア・ファシリティ、周辺装置、SIAM 空き領域など）の管理、ジョブの実行準備、プログラム実行の初期化などをおこなう。各ジョブは順次に行われるジョブ・ステップから構成され、各々のジョブ・ステップは1個のユーザープログラムの実行を要求する。

使用者はジョブ制御文によってジョブ制御プログラムにサービスを要求する。ジョブ制御プログラムは、優先順位と使用可能な資源をもとにしてジョブを自動的にスケジュールし、始動させる。EXEC 7 のもとでは最大 14 個までのジョブの並行処理を管理することができる。ジョブ制御プログラムは、各ジョブの最初のジョブ・ステップの実行前、1つのジョブ・ステップが終了して次のジョブ・ステップへ移る時、およびジョブ全部が終了した時に必要なサービスをおこなう。

ジョブ制御プログラムによっておこなわれるサービスは次のようなものである。

- (1) 入力ジョブ制御ストリームの解析
- (2) 資源の割付け（主記憶装置、周辺機器、再入可能なソフトウェア・ファシリティなど）
- (3) 装置の割当て
- (4) ボリュームとファイル・ラベルの保存、管理
- (5) ジョブ制御手続きの検索とジョブ制御ストリームの拡張
- (6) システム・ファイル・カタログの管理と識別名によるファイルの呼び出し
- (7) チェック・ポイントからのプログラム再開
- (8) オブジェクト・モジュールへのパラメータ入力
- (9) ジョブの自動スケジュールと開始

1.1 制御ストリーム

制御ストリームはジョブ制御言語を用いて書かれた、順序づけられた制御文の集まりである。ジョブは制御ストリームによって定義され、ジョブの処理も制

御ストリームの指示に基づいておこなわれる。制御ストリームには次のものに関する情報が含まれる。

- (1) ジョブの識別とアカウンティング
- (2) 実行されるべきプログラムの名前
- (3) ジョブ・スケジューリング
- (4) カタログ・ファイル
- (5) 装置の割り当て
- (6) ファイル・ラベル
- (7) 主記憶装置の割り当て
- (8) ジョブ・ステップ間の磁気テープ制御
- (9) ジョブの終了

さらに、制御ストリームはジョブの実行に必要とされるデータおよび言語プロセッサに対するソース・コードを含んでいることもある。

各ジョブは EXEC 7 によってそのジョブを識別し制御ストリームの存在場所をつきとめるために特有の名前を持っていなければならない。ジョブで必要とされる各々の入力ファイル、出力ファイルに対し制御文に基づいて装置が割当てられる。

制御ストリームは入力シンビオントによってシステム入力装置から読み込まれ、あとで検索および処理されるためにディスク上のジョブ・ファイルにしまわれる。制御ストリームの中のジョブ制御文は通常順次に処理されるが、一部分を条件付きで飛び越したり、無条件に飛び越して処理することもできるように考慮は行われている。

2. ジョブ管理

ジョブ管理は EXEC 7 の非常に重要な機能の1つであり、使用者によって各ジョブに与えられた緊急度を忠実に反映しつつ、一方ではシステム資源を無駄なく使用するようにジョブの実行を調整するものである。この調整を効果的におこなうために入力シンビオントによってジョブ制御ストリームが処理されて必要な制御用ファイルが作られ、ジョブ・スケジューラは、使用可能なシステム資源を最大限に活用するようにジョブおよびジョブ・ステップのスケジューリングをおこなう。図-1 にジョブの流れとジョブ制御の機

* 沖電気工業（株）ソフトウェア事業部

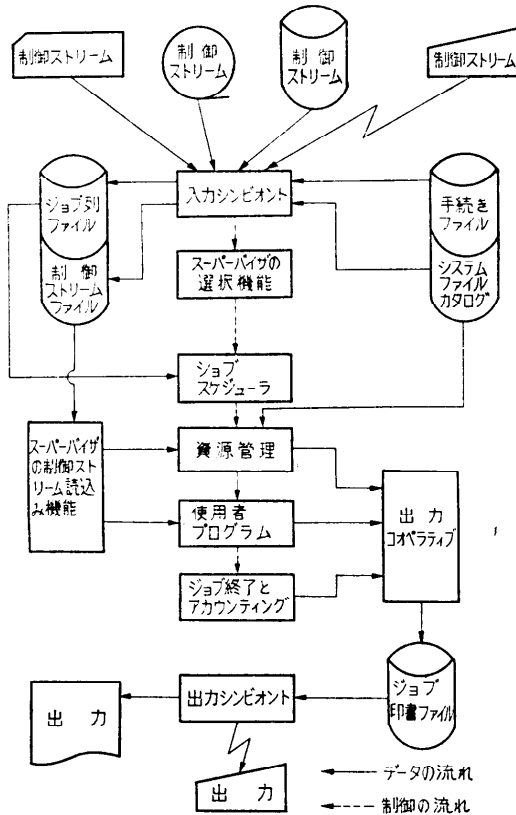


図-1 ジョブの流れ

能を示す。実行されるべき各々のジョブはディスク上の入力作業列に入れられる。複数個の入力制御ストリームから作られたこの列を用いて EXEC 7 はジョブでの優先度と資源の要求をジョブの実行順序に反映させることができる。ジョブ・スケジューラは必要な資源、たとえば主記憶装置、周辺機器などが使用可能であることを確認して個々のジョブ・ステップを選択する。ジョブ・ステップが選択されると制御文の指示にしたがってプログラムが主記憶装置に読み込まれ実行される。

2.1 ジョブの投入

制御ストリームは入力シンボントによって読み込まれ、システムに入力される。入力シンボントは次のような機能を持っている。

- (1) 制御文の読み込み
- (2) 手続きプロセッサの呼び出し
- (3) 呼び出された手続きの拡張
- (4) 制御文の順序と構文法の検査
- (5) 挿入されたデータの分離と貯蔵

- (6) ジョブおよびジョブ・ステップのスケジューリングで使用するための資源要求のまとめ
- (7) ジョブ優先度の認識
- (8) 優先度にしたがってジョブを待行列に入れること
- (9) すでに処理済のジョブの制御ストリームを除去すること

操作員がシステム入力装置に制御ストリームをセットすることにより（たとえばカード読取り装置の場合には‘アテンション’割り込みにより）入力シンボントが自動的に起動させられ制御ストリームが読み込まれる。システム入力装置は、カード読取り装置、磁気テープ装置、磁気ディスク装置、遠隔端末装置いずれも同等に使用可能である。制御ストリーム内のすべての制御文は挿入されたデータ以外のものについてその順序と構文法が検査される。誤りが発見されると制御文と共に適切な診断がジョブの印書ファイルに入れられる。構文法の誤りが発見されると制御ストリームとそのジョブに関する情報はすべて除去されるが、挿入されたデータは使用者の要求により除去せずに残しておくこともできる。手続き呼び出し文が制御ストリームの中に発見されると、要求された手続きが手続きライブラリから検索され、与えられたパラメータを使用して拡張される。拡張された結果の制御文は通常の制御ストリームで与えられた制御文とまったく同様にとりあつかわれる。

2.2 ジョブの選択とスケジューリング

ジョブ・スケジューリングは EXEC 7 により自動的におこなわれるが、基本的には使用者が‘JOB’制御文を用いて各ジョブに与えた優先度を持ちいておこなわれる。システム生成時に各使用者システムにもっとも適合した優先度段階の数、優先度の型、ジョブ選択方法が決定されスケジューリング・テーブルとして準備される。ジョブ・スケジューラはジョブの選択にあたって次のことがらを調べる。

- (1) 使用者によって与えられた各ジョブの優先度段階
 - (2) 各ジョブで必要とする資源の量
 - (3) まだ選択/実行されていないジョブのシステム内での位置関係（投入された前後関係など）
- すべての優先度段階は次の3つの優先度の型に区分される。

- (1) ‘NORMAL’ 優先度

この型がもっとも普通の型であり通常大部分の

ジョブがこの型に属する。この型のジョブは緊急処理を必要としないということで特徴づけられる。‘NORMAL’ 優先度のジョブは使用者の指定によって ‘PREEMPTABLE’ (‘PREEMPTIVE’ 優先度のジョブによって取り消される対象となり得る) あるいはスワップ・アウト可能という宣言をすることができる。

(2) ‘OVERRIDE’ 優先度

これはシステム内で2番目に高い優先度の型である。あるジョブがこの優先度を与えられており、システム内に ‘PREEMPTIVE’ 優先度のジョブがなければこのジョブの遂行に必要なシステム資源をまずこのジョブに割り付けようという試みがなされる。主記憶装置の未使用部分が ‘OVERRIDE’ 優先度のジョブを実行する為に不十分な場合にはスワップ・アウト可能なプログラムが一時抑止されてディスク上にスワップ・アウトされ、それによって空いた主記憶装置が割り当てられる。

(3) ‘PREEMPTIVE’ 優先度

これはシステム内で最高の優先度の型である。この型の優先度のジョブの実行に必要なシステム資源はまずこのジョブに割り付けようという試みられる。もし未使用のシステム資源がこのジョブの実行に不十分な場合 ‘PREEMPTABLE’ と宣言されたジョブの実行が取り消され、そのジョブで使用していた資源が ‘PREEMPTIVE’ 優先度のジョブに割り付けられる。

これらの優先度の型はシステム生成時に決められた優先度段階と結びついてジョブ・スケジューリングに使用される。システム生成時には優先度段階毎にジョブの選択方法も決定される。次のようなジョブの選択方法をとることができる。

(1) ‘FIFO’ (First In, First Initiated)

これはもっとも簡単なジョブ選択方法である。各々の優先度段階毎にシステムに投入された順番にジョブが開始されるというやり方である。したがってあるジョブが必要なシステム資源が使用できないために実行開始できないような場合、同一の優先度段階でそのジョブより後からシステムに投入されたジョブは選択の対象にはならない。

この選択方法は必ずしもシステム資源を最も有効に利用できるとは限らないが、同一優先度

段階であればジョブの実行開始が各ジョブのシステムへの投入順におこなわれることが保障されているという利点がある。

(2) ‘FIFO’ (First In, First Fit)

これは ‘FIFO’ 選択方法の一種の拡張である。待ち行列に入っているジョブは優先度段階毎にシステムへの投入順に選択可能であるかどうかしらべられる。あるジョブが必要なシステム資源が使用できない為に実行開始されないような場合、同じ優先度段階で後から投入されたジョブで現在使用可能なシステム資源の範囲内で実行可能なものがあれば選択される。使用可能なシステム資源の不足のために実行できないジョブは必要な資源が使用可能となるまで待たされる。

この選択方法によればすべてのシステム資源をもっとも有効に使用することができるが、実行のために多くのシステム資源を必要とするようなジョブの実行が遅れがちになるという傾向がある。

EXEC 7 のジョブ・スケジューリングにおける、ジョブの選択方法に対する基本的な考え方は優先度段階の低いジョブでもシステム資源が使用可能である限りは実行させようというものである。ある優先度段階に現在使用可能なシステム資源の範囲内で実行可能なジョブがないような場合には次に低い優先度段階のジョブがしらべられる。各々の優先度段階に対して、使用者はシステム生成時に ‘この優先度段階で同時にいくつのジョブを実行開始して良いか’ を指定する。また、FIFO の選択方法を指定した優先度段階に対しては ‘次の優先度段階をしらべる前に、この優先度段階の中で最大いくつのジョブまでしらべるか’ も指定する。

2.3 ジョブの終了

ジョブの終了には2つの形がある。正常終了と異常終了である。どちらの場合でもジョブ制御プログラムの中のジョブ終了機能によってそのジョブに割り付けられていたシステム資源は他のジョブで使用可能とするために解放される。制御ストリームの中に残っている制御文やデータはすべて無視される。

(1) 正常終了

正常終了とは制御ストリーム内の制御文を順次処理していった、ジョブ終了制御文 (/&) に到達したときにおこなわれるジョブの終了であ

る

(2) 異常終了

異常終了は実行中のプログラム、制御ストリームあるいはシステム操作員指令によって引き起こされる。プログラム実行中の異常終了は次のいずれかの原因によりおこなわれる。

- (i) ジョブ実行に必要な時間の見積りを越えた。
- (ii) プログラムによって 'CANCEL' マクロ命令が実行された。
- (iii) 実行中のプログラムが定められた規則に反する行動をした。

使用者は異常終了の場合には主記憶装置の内容を表示するように制御ストリームを通じて指定することができる。

入力シンピオントによって制御ストリーム内に誤りが発見された場合にはそのジョブは待ち行列には入らず、制御ストリームと診断が印書されジョブは除去される。

ジョブの終了時には次のようなことがおこなわれる。

- (1) ジョブに割り付けられていた装置の解放
- (2) 主記憶装置の解放
- (3) 未処理の磁気テープ制御の開始
- (4) SIAM スクラッチ空き領域の開放
- (5) 制御卓上への終了メッセージの表示
- (6) 終了したジョブに関するアカウント情報の作成
- (7) リンク・パック領域内のソフトウェア・ファシリティの解放
- (8) 未処理の内容表示 (Dump) の開始
- (9) 正常終了の場合にはカタログの世代群登録の更新

2.4 ジョブ印書ファイル

各々のジョブに対して自動的にディスク上に出力シンピオント用のファイルが割り付けられる。このファイルは次のようなものの印書をおこなうために使用される。

- (1) ジョブ制御文
- (2) 診断および警告メッセージ
- (3) 言語プロセッサの出力リスト
- (4) その他 EXEC 7 の構成要素、各種プロセッサ、ユーティリティ・プログラムの出力リスト
- (5) アカウンティング情報

もし使用者が望むならば、使用者プログラムからの出力をこの印書ファイルの中にも含めることができる。

3. 資源管理

ジョブ制御プログラムのなかの資源管理機能によって管理される資源としては、主記憶装置、周辺機器、ソフトウェア・ファシリティ、SIAM 空き領域が含まれる。ソフトウェア・ファシリティとはたとえばデータ管理共用コードのようなもので、通常共用システム資源として扱われる。システム資源は通常各々のジョブに対してそのジョブで必要な程度必要な量だけ割り付けられる。ジョブ毎の資源必要量はそれぞれのジョブによって大巾に異なっていることもあるが、使用者システムによっては多くのジョブが似たような資源必要量をもっている場合も有り得る。EXEC 7 においては通常のジョブ毎の資源必要量にもとずいた動的な資源割り付けばかりではなく、使用者の必要性にもとずいて固定資源区域を設けることもできるようになっている。固定資源区域とは必要なシステム資源群を固定資源区域に割り付けておき、そのシステム資源群を用いて実行できるジョブを順次この固定資源区域を用いて実行させるものである。固定資源区域は操作員がシステムの運転中に自由に設定することができる。

3.1 装置の割当て

周辺装置の割当ては制御ストリーム内の制御文で与えられた情報にもとずいておこなわれる。使用者は装置の割当てに関して次のような情報の全部あるいは一部分を制御文で与えることができる。またこれらの情報をシステム・ファイル・カタログに登録しておくことにより、ファイル識別名を与えることによって装置の割当てを受けることもできる。

(1) 論理的装置指示

使用者は論理的装置番号を示して装置の割当てを要求できる。

(2) ボリューム通し番号

使用者は装置を要求するのにボリューム通し番号を指定できる。もしこのボリュームが装填されている装置があればその装置はジョブに割当てられる。

(3) 装置割り当ての持続期間

使用者は装置の割当てられる持続期間がジョブ単位であるか、ジョブ・ステップ単位であるかを指定できる。

(4) 装置の型の代用

基本となる型の装置が使用不可能の場合に代用として割当て可能な型の装置を指定することができる。これは論理的装置指示の付け方で実現することもできるし、システム生成時に装置の型の代用テーブルを作成しておきそのテーブルを制御文で参照することによっても実現できる

(5) 交換用装置

使用者はジョブ実行中の交換用として同じ型の他の装置を割当てることができる。この機能は複数ボリュームにまたがる大容量の順編成ファイルを処理する時によく使われる。本来の装置が割当てられていなければ交換用装置は割当てられない。

(6) 任意の装置

使用者はある装置の割当てがなくてもジョブの処理は可能であることを指定することができる

(7) 共用可能な装置とボリューム

共用可能な装置あるいはボリュームは同時に複数個のジョブに割当てられることが可能である。この指示を用いるのは通常ディスク・サブシステムである。

(8) ファイルの特徴

使用者はファイル編成の型を指定できる。

3.2 前装填ボリューム

ジョブ・ステップが装置およびボリュームを要求したときジョブ制御プログラムの資源管理機能がそのボリュームがどれかの装置に装填されているかどうかをしらべる。もしそのボリュームがどの装置にも装填されていない場合には使用可能な適当な装置が資源管理機能によって割当てられ、操作員はその装置に必要なボリュームを装填しなければならない。必要なボリュームがすでにいずれかの装置に装填されている場合資源管理機能によって次のいずれかの行動が行われる。

- (1) その装置が共用可能な場合にはその装置は無条件にそのジョブに割当てられる。
- (2) その装置が共用不可能でかつすでに他のジョブに割付られている場合には、この時点では装置の割当てはおこなわれない。
- (3) その装置が共用不可能であるがまだどのジョブにも割当てられていない場合には、その装置がジョブに割当てられる。

ボリュームが前装填されているのは、ボリュームが他のジョブのために装填されていて取りはずされなかった場合と、ボリュームの自動認知機能が実施された

場合である。

3.3 システム・ファイル・カタログ

EXEC 7 のジョブ制御プログラムの機能の1つとしてシステム・ファイル・カタログの保守機能がある。ジョブ制御プログラムはファイルに関する情報をシステム・ファイル・カタログ上で保守することより、システム・ファイルおよび使用者データ・ファイルについての更新状態をいつも知っておくことができる。

また使用者からファイル識別名だけがあたえられたときそのファイルがどこにあるかを知り、使用者プログラムに割付けるために必要な情報を得るためにもシステム・ファイル・カタログが参照される。ファイルが最初に定義され必要な情報をシステム・ファイル・カタログに登録するために 'LFD' 制御文が使われる。また 'LFD' 制御文は必要なカタログ構造を作りあげたりカタログの保守をおこなうためにも使われる。使用者がカタログされているファイルを使用するときには 'CALCAT' 制御文を用いる。システム・ファイル・カタログの機能は次のようなものである。

- (1) ファイル識別名だけが与えられたとき自動的にそのファイルの存在場所を見つけ出すこと。
- (2) データ・ファイルの世代の自動管理
- (3) システム・プロセッサのためのスクラッチ領域の管理
- (4) データ・ファイルの合言葉による保護
- (5) データ・ファイルの共用可能性または独占的使用に関する制御。

3.3.1 特徴

システム・ファイル・カタログの中にはファイルを検索、処理するために必要な情報が登録されているが、使用者は制御ストリームで指定することによってその内容を修正することができる。またその修正がそのジョブの実行中のみの一時的なものであることを指定することもできる。カタログされているファイルのファイル識別名は修飾名を (Qualifier) 持つことができる。使用者は必要に応じて何段階もの修飾を行なうことができる。システム・ファイル・カタログはこれらの修飾名にもとずいて階層構造になっている。ファイル自身はこの階層構造の最低段階に位置しているとみなすことができる。修飾名にもとずいたシステム・ファイル・カタログの構造はジョブ制御文によって構成され、修正される。システム・ファイル・カタログを使用するときの規則は次のようなものである。

- (1) 全システム・ファイル・カタログは通常シス

テム居住ボリューム上に常駐する。

- (2) 副カタログはシステム居住ボリューム以外に置くことができるがそのボリュームは制御ストリーム読み込みおよび資源割当て中は利用可能となっていないなければならない。
- (3) システム・ファイル・カタログは SIAM 領域内に存在する。
- (4) データ・ファイルはあらかじめ定義されたカタログ構造にしたがってカタログされなければならない。

3.3.2 世代群ファイル

単一のファイルが順次更新され発展し世代が変わっていくような場合がある。このような場合のためにシステム・ファイル・カタログは同一ファイルの個々の世代に対して世代番号を割り当てる機能を持っている。この機能によって使用者は世代の異なるファイルすべてを同一名でカタログすることができる。一旦世代ファイルとしてカタログされたファイルに対して使用者はファイル識別名とその絶対世代番号または相対世代番号によって参照することができる。

3.3.3 合言葉による保護

カタログされたファイルは合言葉によって保護することができる。合言葉はシステム・ファイル・カタログの中に登録されておりファイルを読み込むための合言葉と、ファイルに書き込むための合言葉とを別々に登録しておくこともできるようになっている。

3.3.4 ファイルの共用

カタログされているファイルを使用するときに使用者はそのファイルを他のジョブと共用することが可能であるか、あるいは独占的に使用するかを指定することができる。個々のファイルに対して現在そのファイルを共用している使用者数あるいはそのファイルが独占的に使用されているかどうかを示す標識がシステム・ファイル・カタログの中に維持されている。

3.4 SIAM 領域

EXEC 7 においてはすべての構成要素の入出力を統一的に SIAM (System Integrated Access Method) と呼ばれるアクセス方式を通じておこなうようになっている。たとえば言語プロセッサの用いるクラッチ・ファイル、ライブラリアンの管理するプログラム・ライブラリなどである。SIAM を用いて読み書きされるディスク上の領域を 'SIAM 領域' と呼ぶ。SIAM 領域のうちでも非常に頻繁に使われるクラッチ領域は 'システム・クラッチ領域' として特別の資源管

理を受ける。クラッチ領域を必要とする場合使用者は制御ストリームの中でその必要量を宣言すればジョブ制御の資源管理機能により適当な装置上のボリュームからクラッチ領域が割当てられる。

SIAM によって次のような形態の処理をおこなうことができる。

- (1) 順処理 (Sequential File)
- (2) 副ファイルを持つ順処理 (Sequential with Subfile)
- (3) 名前による乱処理 (Random by Name)
- (4) 区分ファイル (Partitoned file)

SIAM の採用により EXEC 7 におけるすべての構成要素の入出力インターフェイスが統一的に管理されるようになると同時に EXEC 7 が必要とする資源を効率よく融通性を持って管理できるようになっている必要に応じて通常の使用プログラムでも SIAM を使用できるようなインターフェイスが準備されている。

4. ジョブ制御言語

ジョブ制御言語は使用者がジョブを定義し EXEC 7 に知らせるための言語であり、一連のジョブ制御文によって記述される。個々のジョブ制御文はジョブの実行、処理を制御するための情報を含んでいるが大別して次の 6 つの群のいずれかに属する。

- (1) ジョブ全体を規定するもの
'JOB', 'ACCT', 'CANCEL' など
- (2) 資源の割り付けを指示するもの
'DVC', 'LFD', 'CALCAT', 'SCRTCH', 'QUAL' など
- (3) ジョブ・ステップでの操作を指示するもの
'EXEC', 'ALTER' など
- (4) プログラム・デバッグ用のもの
'SNAP', 'TRACE' など
- (5) 制御ストリームの処理手順を指示するもの
'SKIP', 'SET', 手続き呼び出し文など
- (6) 装置制御用のもの
'NTC', 'OPR', 'FIXPAR' など

4.1 制御文の形式

制御文は 4 つの欄を持っている。それらの欄は 1 つ以上の空白によって区分される。各欄は次の順番で書かれる。

(1) 指示欄

この欄はデータと制御文を区別するために使われ、//, /&, /*, /\$ のいずれかで始まる。

(2) 操作欄

この欄は実行されるジョブ制御動作、あるいは機能の名前が書かれる。この欄は // ではじまるすべての制御文について必要である。

(3) オペランド欄

ジョブ制御機能が動作をほどこす対象となる項目またはジョブ制御機能が実行される方法についての情報が記述される。// ではじまる制御文で対象あるいは方法を指定する必要のある操作についてのみこの欄を書くことができる。

(4) 注釈欄

この欄には必要に応じて種々の注釈が書かれる。この欄に書かれた内容はジョブ制御プログラムに対して一切意味を持たない。

指示欄がカードの第1欄からはじまること、各々の欄が1つ以上の空白で区別されること、各々の欄の中には特別な場合を除いて空白があってはならないことの規約の範囲内でジョブ制御文はまったくの自由形式である。/\$, /*, /&以外の制御文は1行に複数個の制御文を書くことができる。また一部の制御文は複数の行にまたがって継続して書くこともできる。

オペランド欄には通常1個以上のパラメータがコンマで区切って書かれる。個々のパラメータは定位置パラメータまたは手がかりパラメータである。定位置パラメータは手がかりパラメータより前に置かれ、定められた順序で書かなければならない。定位置パラメータを省略する場合にはそのパラメータがオペランド欄の最後でない限り省略したことを示すコンマを必ず書かなければならない。手がかりパラメータはオペランド欄にどんな順番で書かれても良い。手がかりパラメータは一般に手がかり語の後に等号をつけ、その後に記述部を書くという形式になっている。手がかりパラメータの記述部および定位置パラメータは一連の副パラメータから構成される場合もある。副パラメータは前後を括弧でかこまれ、各々はコンマで区切られた要素のリストである。

4.2 手続きプロセッサ

ジョブ制御手続きプロセッサは1つの手続き呼出し制御文を使うことにより制御ストリーム内に制御文群を生成することができる。ジョブ制御手続きの使用により制御ストリーム作成の労力、時間を大巾に節約することができる。また誤り発生の危険性を減少させることができる。手続きプロセッサは手続き定義にしたがって手続き呼出し制御文を拡張する。手続き定義は

手続きファイルの中にあり EXEC 7 ディスク・ライブラリアンによって作成および修正がおこなわれる。手続き定義はモデル文、手続き指示語、注釈により構成されている。手続き指示語は制御ストリーム内に生成されるべきモデル文の数や順序を決定したり、手続き呼び出し制御文で与えられたパラメータをどう使うかを決定するとき、手続きプロセッサのとるべき行動を指示するためのものである。手続き指示語の中に条件付き分岐指示語('GOIF' 指示語)が含まれている。これは手続き呼出し制御文で与えられたパラメータなどをしらべることによって呼出された時の条件によって生成される制御文の種類、数、順序を種々変えたい場合などに使用される。また条件をしらべのために種々の関係演算子を使用できるようになっている。手続きモデル文は手続きが手続きプロセッサによって拡張された時に制御ストリーム内に生成されるジョブ制御文やデータである。モデル文内にパラメータ表現を用いることが出来る。モデル文が制御ストリーム内に生成される時にはこのパラメータ表現は手続き呼出し文で与えられた対応するパラメータで置きかえられる。このようにジョブ制御手続きは非常に柔軟な運用ができるような考慮がはらわれている。

5. ジョブ制御ストリームの例

カタログされたライブラリ 'LIBIN' から FORT-RANのソース・コードを読み込んでコンパイルし実行する例を掲げる。実行時のデータ・カードは制御スト

```
// JOB FORTRAN,,, C000, 10000
// ACCT 970001
*)// CALCAT SYS#. LIB. S100, FLNM=LIBIN
// SCRTCH 3, M
**)// EXEC FOR
*)// PARAM IN=EXPROG/LIBIN
// CALCAT SYS#. LIB. S19, FLNM=RESV#
// SCRTCH 1
// EXEC LNK
// EXEC EXPROG, MCL
/!$
data cards
.
.
/* . END OF Data
/& . END OF JOB
```

表-1

リームから読み込み、結果はジョブ印書ファイルに出力する。ソース・コードをカードで入力する場合には*印のカードを取り除き、**印のカードの後に /\$, /* で前後をはさんでソース・コードを置けば良い。

(昭和 49 年 5 月 15 日受付)