

JAERI-M  
82-204

一次元トカマク輸送コード・ライブラリ・  
システム (LIBJT60) …… 1  
(使用手引)

1982年12月

JT-60実験解析コード作業グループ

(編) 平山 俊雄

日本原子力研究所  
Japan Atomic Energy Research Institute

JAERI-M レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村 日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Section, Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1982

---

編集兼発行 日本原子力研究所  
印刷 日立高速印刷株式会社

一次元トカマク輸送コード・ライブラリ・システム  
(LIBJT60) …… 1  
(使用手引)

日本原子力研究所東海研究所大型トカマク開発部  
JT-60 実験解析コード作業グループ\*  
(編)平山 俊雄

(1982年12月10日受理)

大規模なソース・プログラムを管理するライブラリ・システムを開発した。本システムは、大規模プログラムのデータ構造、およびプログラム構造を、柔軟性に富んだ構造に変換し、ユーザの多種多様な仕様に対応することを目的としている。ライブラリ・システムを、1次元トカマク輸送コードに適用することにより、汎用トカマクコードは、小型で、機動性の高い単機能コード群に分離することができた。また、本システムでは、コード作成時のユーザの負担を軽減するために、システム専用の編集支援ツールを新たに開発した。ユーザは、ライブラリ・システムの特性を踏まえたコードを作成することにより、発展性の高いコードを効率よく開発することができる。

---

\* 平山俊雄, 谷 啓二, 安積正史, 滝塚知典(核融合研究部), 中西雅裕, 折居茂夫(東海大学), 吉田英俊, 二宮博正, 関 省吾, 下村安夫

Library System for a One Dimensional Tokamak  
Transport Code ( LIBJT60 ) ----- 1  
(Handling Manual)

(Ed.) Toshio HIRAYAMA

Working Group for Developing Experimental Analysis Code<sup>\*</sup>

Division of Large Tokamak Development,  
Tokai Research Establishment, JAERI  
( Received December 10, 1982)

A library system is developed to control and manage huge programs in terms of FORTRAN source. It is applied to widely used one dimensional tokamak transport codes ( LIBJT60 ), which have been developed in the Division of Large Tokamak Development. The structure of data and program in the transport code turn out to be flexible enough to respond to various demands and this gigantic code frame work can be decomposed into groups of a compact code with a specific function. Some editing support tools for programming and debugging are also developed to save programming work. By applying this library system, users can obtain a code whose functions can be efficiently developed.

Keywords; Library System, Tokamak, Simulation, One Dimensional  
Transport Code, LIBJT60, CDGEN, ORG, EOS, STAN, MSTAN

---

\* Toshio HIRAYAMA, Keiji TANI, Masafumi AZUMI, Tomonori TAKIZUKA(Div. of Thermonuclear Fusion Research), Masahiro NAKANISHI, Shigeo ORII (The University of Tokai), Hidetoshi YOSHIDA, Hiromasa NINOMIYA, Shogo SEKI, Yasuo SHIMOMURA

## 目 次

1. 序 .....	1
2. 概 説 .....	2
2.1 ライブラリ・システムの機能 .....	2
2.2 ライブラリ・システムの構成 .....	2
2.3 システム・ファイル .....	4
2.3.1 SYSGEN 記述 .....	5
2.3.2 DATA 記述 .....	8
2.3.3 PROGRAM 記述 .....	10
2.4 処理プログラム .....	11
2.4.1 EOS および EOS 言語 .....	11
2.4.2 ORG .....	13
2.4.3 CDGEN .....	14
3. ユーザ DATA SET の作成 .....	17
3.1 モジュール #SYSGEN .....	18
3.1.1 コンパイル時オプションの指定 .....	18
3.1.2 SYSGEN 記述規約 .....	20
3.1.3 #SYSGEN 作成例 .....	21
3.2 モジュール #SIZE .....	23
3.2.1 SIZE データの指定 .....	23
3.2.2 SIZE 記述規約 .....	24
3.2.3 #SIZE 作成例 .....	24
3.3 モジュール #COMDECK .....	25
3.3.1 COMDECK 記述規約 .....	25
3.3.2 #COMDECK 作成例 .....	26
3.4 ユーザ・ソースモジュールの作成 .....	27
3.4.1 ソース・モジュール記述規約 .....	27
4. データ構造の組上げ .....	32
4.1 データ構造組上げ用 'JCL' の作成 .....	32
4.2 データ構造組上げの実行例 .....	36
5. プログラム構造の組上げ .....	41
5.1 EOS 実行パラメータ .....	41
5.2 ORG 実行パラメータ .....	42
5.3 プログラム構造組上げの実行例 .....	46

6.	JCL, TSS マクロコマンド .....	51
6.1	JCL の作成 .....	51
6.1.1	'LIBJCL' マクロコマンド .....	51
6.1.2	'LIBJCL' の使用例 .....	57
6.2	端末上でのロードモジュール作成 .....	62
6.2.1	TSS コマンド 'LB' .....	62
6.2.2	LB の使用例 .....	62
7.	ユーティリティ .....	70
7.1	PRINT ユーティリティ .....	70
7.2	MSTAN ユーティリティ .....	75
7.3	STAN ユーティリティ .....	77
8.	システム使用例 .....	83
9.	結 言 .....	96
	謝 辞 .....	96
	文 献 .....	96

Contents

1.	Introduction	-----	1
2.	Out line of Library System	-----	2
2.1	Function of Library System	-----	2
2.2	Constitution of Library System	-----	2
2.3	System files	-----	4
2.3.1	Description of SYSGEN	-----	5
2.3.2	Description of DATA	-----	8
2.3.3	Description of PROGRAM	-----	10
2.4	Preprocessor programs	-----	11
2.4.1	EOS and EOS language	-----	11
2.4.2	ORG	-----	13
2.4.3	CDGEN	-----	14
3.	Production of user data sets	-----	17
3.1	#SYSGEN module	-----	18
3.1.1	Option flags for compilation	-----	18
3.1.2	Rule of SYSGEN description	-----	20
3.1.3	Example of make of #SYSGEN	-----	21
3.2	#SIZE module	-----	23
3.2.1	Input of size data	-----	23
3.2.2	Rule of SIZE description	-----	24
3.2.3	Example of make of #SIZE	-----	24
3.3	#COMDECK module	-----	25
3.3.1	Rule of COMDECK description	-----	25
3.3.2	Example of make of #COMDECK	-----	26
3.4	Production of user source modules	-----	27
3.4.1	Rule of PROGRAM description	-----	27

4.	Construction of data structure-----	3 2
4.1	JCL procedures for construction -----	3 2
4.2	Example of construction of data structure-----	3 6
5.	Construction of program structure-----	4 1
5.1	Practicing parameters for EOS preprocessor-----	4 1
5.2	Practicing parameters for ORG preprocessor-----	4 2
5.3	Example of construction of program structure-----	4 6
6.	Macro commands for JCL and TSS-----	5 1
6.1	JCL procedures -----	5 1
6.1.1	'LIBJCL' macro command -----	5 1
6.1.2	Example of use of 'LIBJCL' -----	5 7
6.2	TSS procedures -----	6 2
6.2.1	'LB' TSS command -----	6 2
6.2.2	Example of use of 'LB' -----	6 2
7.	Utility programs -----	7 0
7.1	PRINT utility -----	7 0
7.2	MSTAN utility -----	7 5
7.3	STAN utility -----	7 7
8.	Example of use of Library System-----	8 3
9.	Summary -----	9 6
	Acknowledgements -----	9 6
	References -----	9 6

## 1. 序

核融合研究開発の進展にともない、トカマク・プラズマの計算機シミュレーション・コードの規模は、巨大化の一途をたどっている。汎用性を備えた巨大コードは、単一機能のコードとして使用する場合にも、必要とする記憶領域は非常に大きく、コードを動かす上でも、また開発する上でも、その巨大性は大きな障害となっている。コードの巨大化は、プログラムの維持管理を困難にすると共に、仕様変更によるプログラムの修正、改良をむずかしくしている。その結果、仕様変更のたびに、類似した機能のコードを重複して開発するなどの事態が生じている。さらに、類似コードによる計算結果を比較検討する場合でも、計算アルゴリズムなどの違いのために、容易におこなえないのが現状である。また、コード開発に相当のコストがかかるようになってきているため、コードの開発自体を効率よく進めることも必要である。

一次元トカマク輸送コード・ライブラリ・システム（以後、LIBJT 60 と呼ぶ）は、以上のような状況を打開するために作成された、ソースプログラムのライブラリ・システムである。本システムは、SSLなどのソースライブラリと異なり、制御構造のみでなくデータ構造に関しても、ユーザの要求に対応できる柔軟性を備えている。また、サブプログラム相互の呼出し関係を、各プログラムの直後に書込むことによって、必要なルーチンのみが目的コードとして組上げられるようになってきている。開発の目標は、以上の諸点にある。

### (1) サブプログラムの共有化

シミュレーションコードは、単機能のサブプログラムを多数含んでいる。各種の目的を持ったコードをこのシステムから作成する場合に、これらの単機能ルーチンを共通に使用することにより、開発コストを下げると共に、各々のコードによる計算結果の比較検討を容易にする。

### (2) ロードモジュールの小型化

単機能の計算コードを組上げる際に、不要なサブプログラムとデータエリアを排除することで、所定の機能のみを持った小さなロードモジュールが得られる。これにより、テスト、ランニング等の際のスループットを向上させる。

### (3) ソースプログラムの維持管理

複数の利用者による無秩序なプログラムの改変を防ぐため、ソースプログラムの管理を一元化する。一般利用者はシステム提供のファイルを更新出来ないが、利用者独自のサブプログラムとの結合は容易に行えるようにインターフェースを整備する。

以上の特性によって、データの受渡しにCOMMONを使用する大規模コード全体をソース・ライブラリ化することができた。本システムを、原研で開発されたトカマク解析コード<sup>1)</sup>に適用することにより、トカマク・プラズマの挙動を解析するシミュレーション・コード、および実験解析コードを効率良く組上げる事が可能になった。

以下の節において、ライブラリ・システムの機能、構成およびシステムの使い方について述べる。ライブラリ・システムに組込まれているトカマク解析コードの内容については、ここでは触れない。

## 2. 概 説

### 2.1 ライブラリ・システムの機能

本システムは、ユーザの多様な仕様に対応できるよう、柔軟性に富んだデータ構造、およびプログラム構造を有している。この柔軟構造によって、サブルーチンの共有化、置換の容易性が可能となり、大型汎用コードから、ユーザが必要とする部分のみを、単機能コードとして作り出すことができる。

コードが大型化すると、その編集作業もユーザにとって重い負担となってくる。ライブラリ・システムは、ユーザの負担を軽減するために、編集支援ツールを2種類用意している。その1つは、インクルード・モジュールのツリー情報、および未定義ルーチンなどの診断情報を提供する。他の1つは、FORTRANのソースプログラムを静的に走査して、ルーチンの呼出し関係や、データの相互関連を調べ、プログラムの現状を正確に反映したドキュメントを提供するものである。

本ライブラリシステムに含まれたトカマクプラズマ解析コードは1981年末までに原研「JT60計画室」で開発、更新が行われてきたトカマクプラズマ一次元モデルのシミュレーションコード<sup>1)</sup>を母体として作成された。本システムを使用する事によって、母体となったシミュレーションコードが持っていたすべての機能について、対応する単機能コードを作り出すことが出来る他、同一の計算ルーチンを使った実験シミュレーションコード、実験解析コードを組上げることが出来る。本システムでは、シミュレーション・コード群を総称してWSC (Wide used Simulation Code)と呼び、実験シミュレーション・コードをESC (Experimental Simulation Code)、実験解析コードをEAC (Experimental Analysis Code)と呼ぶ。ここで、ESCはシミュレーション内の基本変数の一部を、実験値を使った外部テーブルに置換した実験データ解析用のコードである。EACは実験データから熱伝導係数、および拡散係数などの内部パラメータを推定するコードである。この内ESC、EACについては将来の拡張とする。

### 2.2 ライブラリ・システムの構成

一次元トカマクコードライブラリシステムはシステムファイルと呼ばれる1本のGEMファイルと、EOS、ORGおよびCDGENの3つの処理プログラムから構成されている。

システムファイルには3種類のモジュール群が含まれており、各々SYSGEN記述、DATA記述、PROGRAM記述と呼ばれる。ユーザは自分の作成したい計算コードに合わせてSYSGEN記述を作成し、処理プログラムに入力するのみで、システムの提供する範囲内において希望する機能を持った計算コードを手にすることが出来る。ユーザが独自のデータエリアまたはサブプログラムを使用したい場合は、別にUSER GEM FILEにDATA記述、PROGRAM

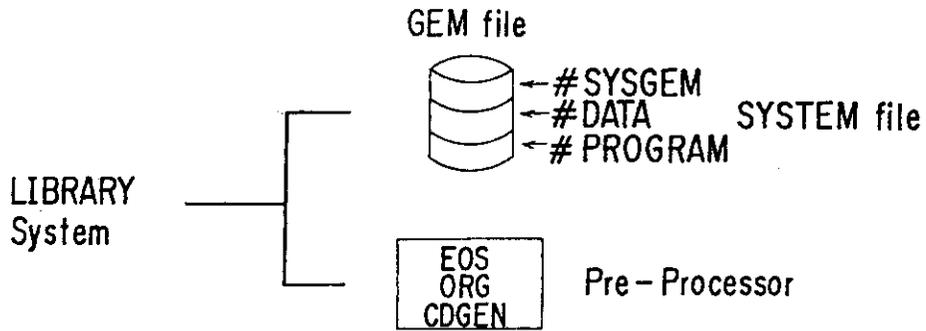


図1 ライブラリシステム構成図

記述を作って処理プログラムに入力する。システム提供のものとの置換，追加は容易な構造となっている。

ライブラリシステムのユーザーは大略次の手順で自分の目標とするロードモジュールを手に入れることができる。

1. SYSGEN記述を書く（目的コードの外部仕様）
2. 追加，置換のためのユーザールーチン等を作る。
3. ロードモジュールのための領域を確保する。
4. JCL作製のマクロ・コマンドを起動する。
5. Submit

この時，データ構造を記述したPSファイルも別に登録される。

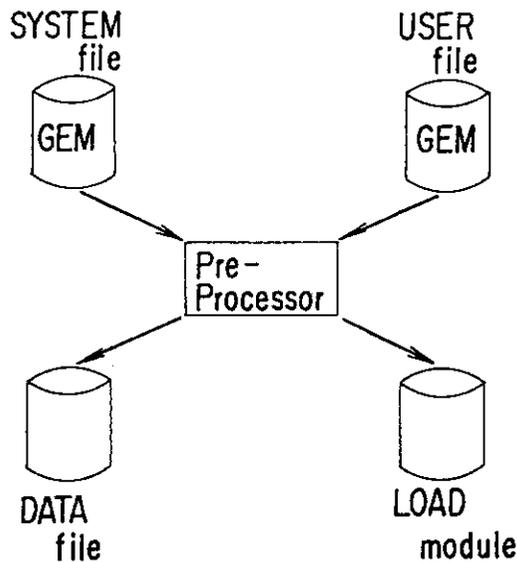


図2 ユーザ・ロードモジュールの作成

ライブラリ・システムは「GEM」および「EOS」と呼ばれる既存のファイル処理システムに依存して作られている。

《 GEM 》

本ライブラリシステムは、FACOM OS M/F4下で供給されている「GEM管理システム」を利用して作られており、特にGEMモジュールの展開機能に大きく依存している。<sup>2)</sup>

《 EOS 》

本システムのDATA記述およびPROGRAM記述は、原則としてEOS言語<sup>3)</sup>で書かれている。EOS言語は原研「理論解析研究室」で開発されたFORTRANに似た言語で、EOSプリプロセッサによってFORTRANソースに変換することが出来る。

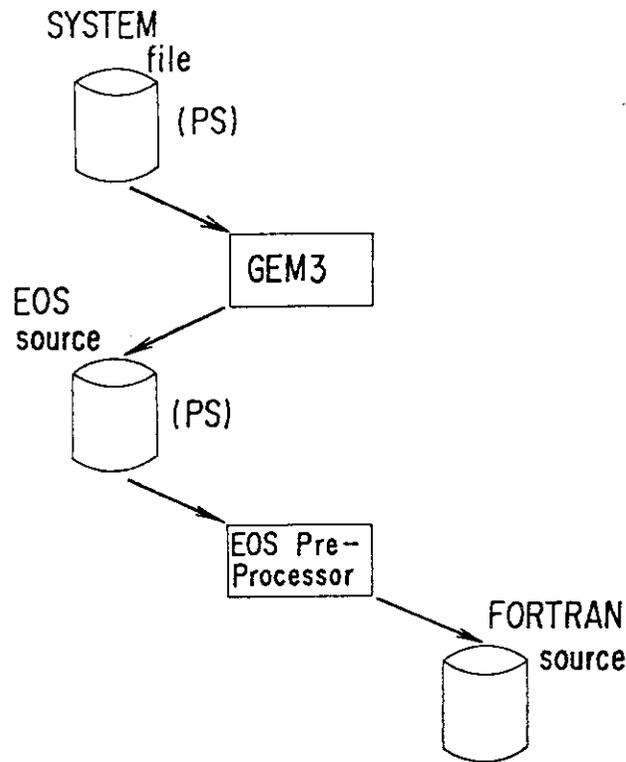


図3 GEMシステムおよびEOSプリプロセッサの利用

2.3 システム・ファイル

ここでは、システムを構成している各種プログラムの役割について説明する。

システムファイルはライブラリ・システムの提供するGEMファイルで、中に含まれるモジュール群はSYSGEN記述モジュール、DATA記述モジュール、およびプログラム記述モジュールの3種に分類される。

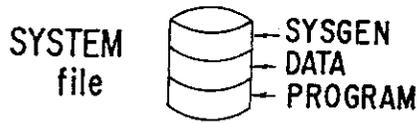


図4 システムファイルの構成

全てのモジュールは次の形式で書かれる。

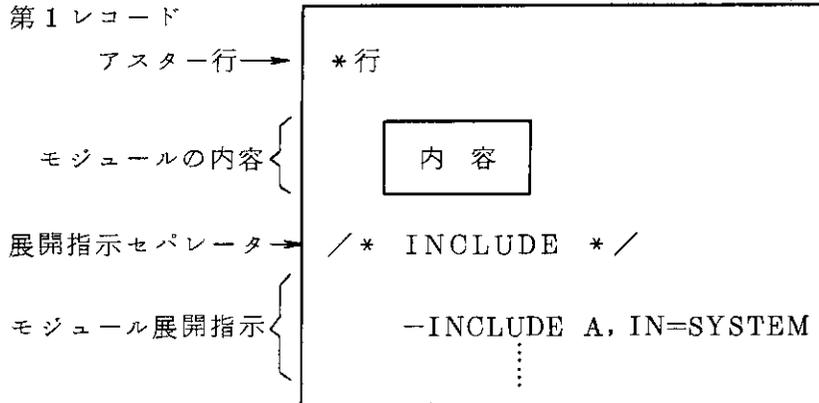


図5 モジュールの記述形式

第1レコードはモジュールの種類および名前を示す。モジュール内容の記述形式はその種類によって異なる。

展開指示セパレータは以後の記述が、ライブラリ内のGEMモジュールを展開する指示であることを示す。詳細は「3. ユーザDATA SETの作成」で述べられる。

### 2.3.1 SYSGEN記述

SYSGEN (System Generator) 記述は、目的コードの外部仕様を表現するモジュール(#SYSGEN)であり、ユーザは必ず記述しなければならない。システムファイルには、その一例('J2855.LIBJJ60.FORT(#SYSGEN)')が組込まれているので、ユーザは自分のGEMファイルにこれをコピーして書換えることによって、#SYSGENの変更を容易におこなうことができる。

SYSGEN記述モジュールは下の様な形式で書かれる。

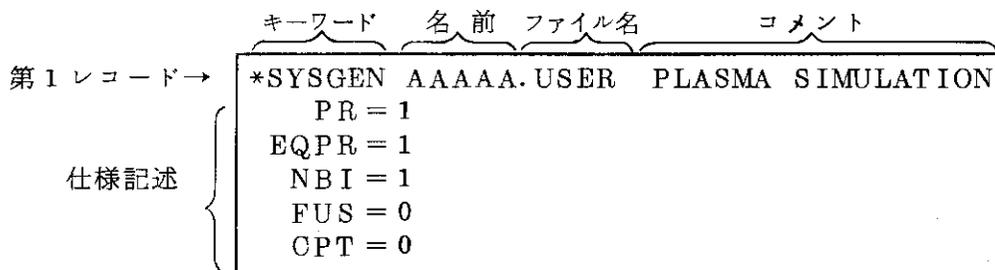


図6 SYSGEN記述の例

第1レコードの「キーワード」は第1カラムから左詰めで書かれる。「名前」は第2レコード以下の仕様記述群の名称、「ファイル名」はこのデータが入っているファイルの名前、また「コメント」は目的コードの標題である。

仕様記述は対象とするプラズマの種類，解かれる基礎式の種類，解法等をパラメータ形式で記述したもので4つの段階に分けられる。

表1 仕様記述の4段階

STEP 1	プラズマ粒子の選択
STEP 2	基礎式の選択
STEP 3	項の選択
STEP 4	解法の選択

ユーザは各段階に対応して，きめられたパラメータに値を代入し目的とするコードの外部仕様を指定する。また予約名以外のパラメータを任意に定義して使用することも出来る。以下にパラメータを示す。

表2-1 STEP 1 プラズマ粒子の選択

パラメータ名	値	意味
PR	0	Proton が { 存在しない 存在する
	1	
DE	0	Deuteron 同上
	1	
TR	0	Tritium 同上
	1	
HE	0	Helium 同上
	1	
CA	0	Carbon 同上
	1	
OX	0	Oxygen 同上
	1	
FE	0	Iron 同上
	1	
AR	0	Argon 同上
	1	
MO	0	Molibdenum 同上
	1	

表 2-2 STEP 2 基本式の種類

パラメータ名	値	意 味
EQPR	0 1	Proton の連続の式を { 解かない 解く
EQDE	0 1	Deutron "
EQTR	0 1	Tritium "
EQHE1	0 1	He <sup>+</sup> イオン "
EQHE2	0 1	He <sup>++</sup> イオン "
EQK	0 1	不純物イオン "
EQCZ	0 1	電流の式
EQBP	0 1	ポロイダル磁場の式
EQTEL	0 1	電子温度の式
EQTPR	0 1	Proton 温度の式
EQTDE	0 1	Deutron 温度の式
EQTTR	0 1	Tritium 温度の式
EQTK	0 1	不純物温度の式
EQTI	0 1	バルクイオン平均温度の式
EQEL	0 1	電子密度の連続の式を { 解く 解かない
EQNHY	0 ~ 2	水素の連続の式の数
EQNTHY	0 ~ 2	水素の温度の式の数

表 2-3 STEP 3 項の種類

パラメータ名	値	意 味
FUS	0 1	水素イオン 2 成分の時 { fusion 項なし fusion 項あり
NBI	0 1 2	NBI 項なし 水素 1 に対して NBI 項有り 水素 2 に対して NBI 項有り

表 2-4 STEP 4 解法の選択

パラメータ名	値	意味
CPN	0	水素イオンが2種の時に個別に解く
	1	" 連立させて解く
CPHE	0	Heliumの連続の式を個別に解く
	1	" 連立させて解く
CPT	1	$T_e, T_1$
	2	$T_e - T_1$
	3	$T_e, T_1, T_2$
	4	$T_e, T_1 - T_2$
	5	$T_e - T_1 - T_2$
	6	$T_e, T_i$
	7	$T_e - T_i$

2.3.2 DATA 記述

DATA記述は、目的コードの中で使用されるデータ領域を記述するモジュール群であり、  
# SIZE および # COMDECK の2つのモジュールからなる。

# SIZEには、EOSソース中で使われるサイズ変数の値が記入される。ただし、この中には、  
SYSGEN記述に依存して定まるサイズ変数は含まれない。

```

*SIZE BULK. SYSTEM
MSH=30
MPOINT=10
    
```

図7 # SIZEの例

# COMDECKにはEOSソースのCOMDECK (COMMON DECK)ブロック群が書込まれる。

```

*COMDECK SPACE. SYSTEM
COMMON /SPACE /R (MSH)
*COMDECK CURTAB. SYSTEM
COMMON /CURTAB /CTAB (MPOINT)
    
```

図8 # COMDECKの例

# COMDECKでは、# SIZEで記述されたサイズ変数以外に、SYSGEN記述に対応して

一意的に定まるサイズ変数も使用することができる。このサイズ変数は、コモン変数を作り出す処理プログラム、CDGEN (Common Deck Generator) がSYSGENデータに基づいて定義する変数であり、以下の種類がある。

表3 CDGENで定義されるサイズ変数

サイズ変数名	値又は定義式	
N I O N	N B + N K	PLASMAイオンの種類
N B	N H Y + N H E	B U L Kイオンの種類
N K	N C A   N O X   N A R   N F E	不純物イオンの種類
N H Y	0 ~ 2	水素イオンの種類
N H E	2	Heイオンの電荷数
N C A	6	Carbon "
N O X	8	酸素 "
N A R	1 8	アルゴン "
N F E	2 6	鉄 "
N E U T	N B 0 + N K 0	中性粒子の種類
N B 0	0 ~ 3	B U L K中性粒子の種類
N K 0	0 ~ 1	不純物 " "
N H Y 0	0 ~ 2	水素 " "
N E Q U	1 ~ 9	基礎式の数
N E Q B	1 ~ 4	B U L K連続の式の数
N E Q T B	0 ~ 2	B U L K温度の式の数
N E Q H Y	0 ~ 2	水素連続の式の数
N E Q H E	0 ~ 2	He 連続の式の数
N E Q T H Y	0 ~ 2	水素温度の式の数

○ ユーザによるDATA記述の変更

システムファイル中のDATA記述をユーザーが直接書換えることはできないが、ユーザーは自分のGEMファイル中に作ったDATA記述を、システムのDATA記述モジュールと置換えることができる。また、新たなDATA記述を付加えることもできる。

追加、置換は「\*ブロック」(\*行から次の\*行の直前まで)を単位として行われる。ユーザーの作ったDATA記述中に、システムのDATA記述と同じ名前の\*ブロックがあればユーザーのものが優先的に採られ、無ければ単に追加される。

```
*SIZE BULK. SYSTEM
    MSH = 30
*COMDECK AAA. SYSTEM
    COMMON/AAA/X(MSU)
```

図9 システムファイルのデータ記述

```
*SIZE BULK. USER
    MSH = 70
*COMDECK AAA. USER
    COMMON/AAA/Y(MSH)
*COMDECK BBB. USER
    COMMON/BBB/Z(MSH)
```

図10 ユーザーのデータ記述

上の例では「\*SIZE BULK. SYSTEM」と「\*COMDECK AAA. SYSTEM」は USERのDATA記述で置換えられ、さらに「\*COMDECK BBB」が追加される。

### 2.3.3 PROGRAM記述

PROGRAM記述は、EOSソースのDECKブロックが書かれるモジュールであって、原則として1モジュールに1つの\*DECKブロックが書かれる。システムファイルのPROGRAM記述は、そのモジュール内のルーチンで呼出しているルーチンに対する展開指定を末尾に含んでいる。これによって、例えばMAINルーチンのみを指定すると、SYSGENデータに基づく必要なモジュールがすべてロードモジュールに組込まれ、かつ不要なモジュールは入って来ない。

```
*DECK PPP. SYSTEM
    SUBROUTINE PPP(X)
    X = 0
    Y = 1
    CALL QQQ(Y)
    X = X + Y
    RETURN
    END
/* INCLUDE */
    -INC QQQ, IN = SYSTEM
```

展開指定 →

図11 PROGRAM記述例

上の例では、SYSTEMという機番名に割付けられたシステムGEMファイルからモジュールQQQを取出して、ソース中に展開することが指定されている。

ユーザーによる置換えの方法はDATA記述の場合と全く同じである。

## 2.4 処理プログラム

本ライブラリ・システムは、システム・ファイル、およびユーザのGEMファイルより、目標のコードを作り出す作業を行う処理プログラムとして、EOS、ORGおよびCDGENのプログラムを持っている。

ライブラリ・システムが起動されると、第1段階では目的とするコードのEOSソースプログラムが出力される。この時モジュール相互の支配関係を認識して、モジュールを選び出す役割を受持つ処理プログラムをORG (ORGANIZATION PROCESSOR) と呼び、SYSGENデータを解析して適切なサイズブロックを作り出すプログラムがCDGEN (COMDECK GENERATOR) である。第2段階ではEOSソースプログラムがFORTRANソースプログラムに変換される。

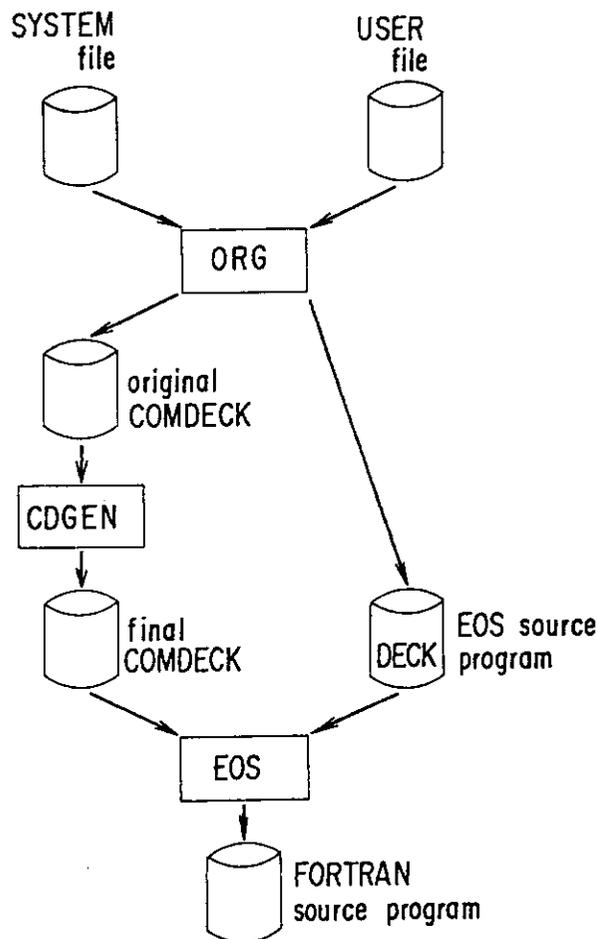


図 1 2 処理プログラムの役割

以下に、これら処理プログラムの機能について説明する。

### 2.4.1 EOSおよびEOS言語

FORTRAN を使用してプログラムの作成、修正を行う際にCOMMON変数などに変更が必

要となる場合がしばしばある。同一ラベルのすべてのCOMMON宣言文を探し出して修正を行う作業は、常に繁雑であり、変更の誤りなどを考えると、プログラム開発の能率の低下を招く。これらの事を解決する手段として生み出されたのが、原研核融合研究部理論解析研究室で開発されたEOSシステム、EOS言語<sup>3)</sup>である。本ライブラリシステムでも、このEOS言語を使用している。

EOSシステムは、以上の目的で作られたFORTRANプログラムのための前処理システムであり、EOS言語が使用されている限りEOS処理して始めて、FORTRANソースプログラムとなる点に注意しなければならない。EOSの機能についての詳細な説明は文献(3)を参照のこと。ここでは簡単な例について説明する。

EOSはサイズ変数定義ブロック(サイズブロック)、展開テキスト定義ブロック(COMDECKブロック)および展開命令を含むテキスト本体(DECKブロック)の3種のブロックから成立っている。

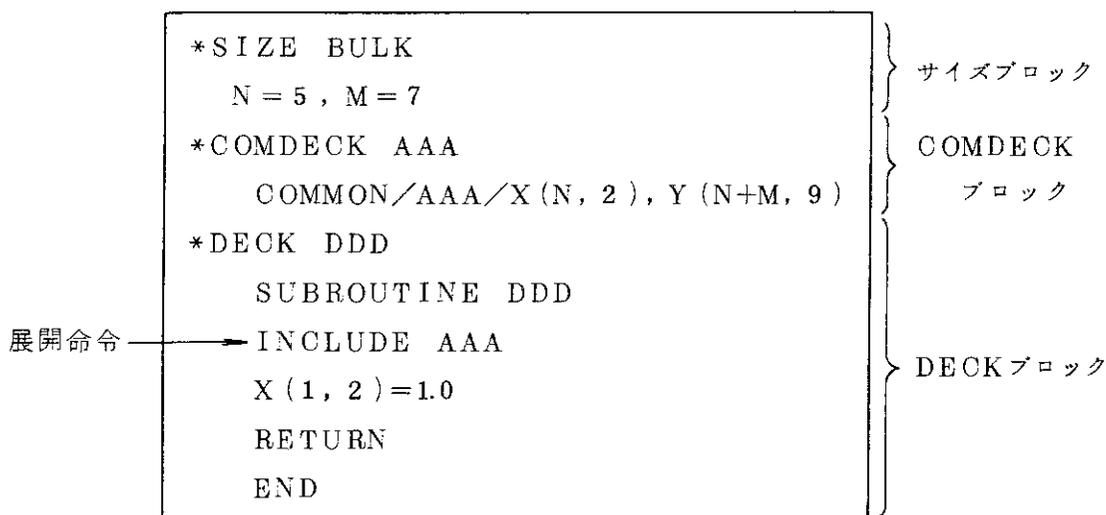


図13 EOSソースプログラム

上の例では COMDECK AAA 中の配列 X, Y の寸法が数字におきかえられて、DECK DDD の 2 行目に展開される。すなわち、EOS プリプロセッサを通ることにより、下の FORTRAN サブルーチンが出力される。

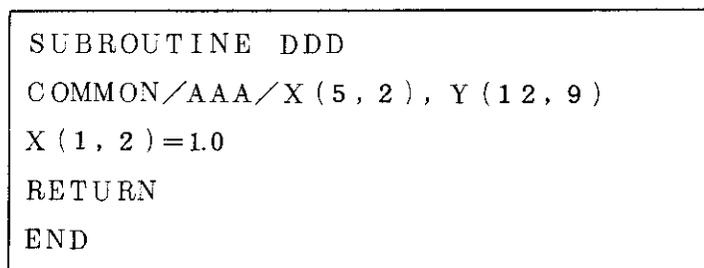


図14 FORTRANソースプログラム

## 2.4.2 ORG (ORGANIZATION PROCESSOR)

ORGの機能は、システム記述（#SYSGENデータ）およびソースライブラリ内に埋め込まれたプログラム構造記述（INCLUDE文など）に従って、ソースプログラムを組み上げるものである。ユーザは、簡単な規約に従って、システム記述、データ構造記述（CDGEN参照）、プログラム構造記述をするだけで、容易にソースライブラリシステムから目的とするソースプログラムを得ることができる。現在のバージョンでは、M-200、GEMシステムにおけるモジュールを扱っている。ORGの機能は次の3つである。

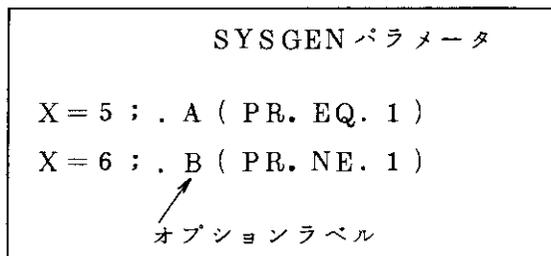
- (1) オプション行の削除
- (2) 展開指定を解析してGEMモジュールを選び出す。
- (3) PRINT文をFORTRANに変換する。

以下に各々の機能について簡単に説明する。詳細については、ORGマニュアル<sup>4)</sup>を参照のこと。

### (1) オプション行の削除

システムファイル中の全てのモジュールのテキストには、行を単位として、ある条件の下にその行を削除する指定を行うことができる。指定の付いている行をオプション行と呼ぶ。

ORGはオプション行の論理値を計算し、'TRUE'以外の時はこの行をテキストから削除する。（3.1.2 SYSGEN記述規約参照）



上の例で、PR=1と#SYSGEN.DATAに記述されていると、Aの行だけが出力ソース中に残ることになる。

### (2) 展開指定によるモジュールの選択

システムファイルの全てのモジュールは、モジュールテキスト本文の下に展開指定セパレータで区切って他のモジュールの展開指定を書くことが出来る。

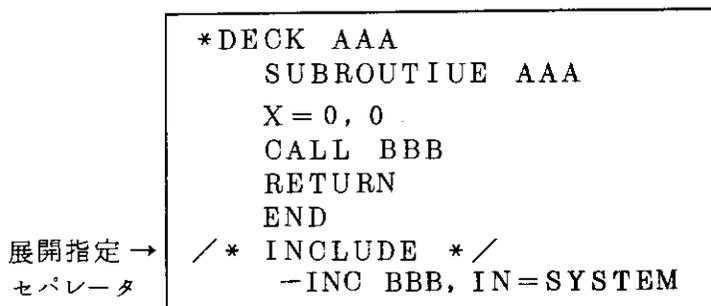


図15 展開指定

ORGは展開指定を解析してモジュール名'BBB'を、ファイル機番'SYSTEM'に割付けられたGEMファイルから取出して、出力ファイル中に展開する。

### (3) PRINT文のFORTRANへの変換

PRINT文は簡単なキーワードとファイル出力項目を並べた文であつて、ORGはこれをFORTRAN文に変換して出力する。

詳細については、7.ユーティリティの使用法を参照。

### 2.4.3 CDGEN (COMDECK GENERATOR)<sup>5)</sup>

CDGENは、ライブラリシステムのユーザが、自分専用のEOSソースプログラムを作る際に、適切なデータ構造を反映したCOMDECKブロック群と、サイズブロックの一部を生成するためのツールである。

CDGENの主要な機能は、次の2つである。

1. データエリアの大きさを決めるサイズ変数の値を計算し、サイズブロックに書加える。
2. データ構造に対応したローカル変数名を生成し、新たなCOMDECKブロックとして追加する。

CDGENはSYSGENデータを解析して、イオン種の数、核種の数、式の数等を計算し、ユーザ専用コードの中で使われるCOMMON変数のサイズを決定する。これにより、データ領域として必要かつ十分な大きさが確保されることになる。

CDGENは上記の主機能に加えて、下記に示すDECKブロック(プログラム本体部分の一部)を生成する機能も備えている。これらのDECKは、データ構造のみに依存して確定してしまいうルーチンとブロックデータであり、主出力である最終COMDECKに対応して作られる。

1. COMMON領域の初期化用ブロックデータ
2. リスタートファイルの入出力ルーチン
3. COMMONダンプルーチン

以下に、簡単な説明を行う。

#### (1) サイズブロックの決定

本コードで使用されているCOMMON配列の中には、対象とするプラズマの成分、基礎式の数等によってその寸法が決められるものが多数存在する。ユーザの入力する外部仕様(SYSGEN記述)に従って、CDGENはこれらの配列の値を計算する。

```
*SYSGEN UUUU. USER   BULK, IMP = OX
      PR = 1
      OX = 1
```

図16 SYSGENデータ例

上のSYSGEN記述に対応して、次のサイズ変数が計算され出力ファイルに書かれる。

```

NB = 1
NK = 8
NION = 9
NB0 = 1
NK0 = 1
NEUT = 2
NHY0 = 1
    
```

図17 CDGENにより定義されたサイズ変数

ここで定義されるサイズ変数については、変数名のままDATA記述中で使用することができる。

(2) ローカル変数名の生成

システム・ファイルのCOMMON配列の中で、SYSGEN依存のサイズ変数(表2-1参照)を寸法に持つ配列名(例えば、DF(MSH, NION): 拡散係数)に対し、決められた文字列を加えて新変数名(ローカル変数名)を作り、元の変数とのEQUIVALENCE文を生成する。生成されたEQUIVALENCE文は、新しいCOMDECKとして登録される。例として、以下のSYSGENデータ(図18)と、COMDECK(図19)の入力を考える。

```

*SYSGEN LIB1D. USER
DE = 1
TR = 1
HE = 1
    
```

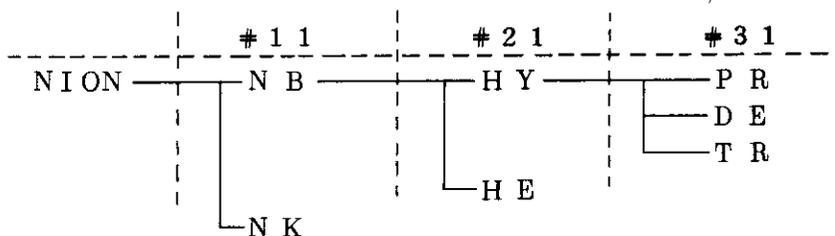
図18 SYSGENデータ例

```

*COMDECK DIFFUS. SYSTEM
COMMON/DIFFUS/DF(MSH, NION)
    
```

図19 #COMDECK例

これに対して、システムのイオン核種の階層構造は以下のようになっているので、



CDGENは、この階層構造に従って新変数名を作成し、次の新しいCOMDECK群を追加する。

```
*COMDECK DIFFUS#11
  DIMENSION DFB(MSH, NB)
  EQUIVALENCE (DF(1, 1), DFB)
```

```
*COMDECK DIFFUS#21
  DIMENSION DFHY(MSH, 2), DFHE(MSH, 2)
  EQUIVALENCE (DFB(1, 1), DFHY), (DFB(1, 3), DFHE)
```

```
*COMDECK DIFFUS#31
  DIMENSION DFDE(MSH), DFTR(MSH)
  EQUIVALENCE (DFHY(1, 1), DFDE), (DFHY(1, 2), DFTR)
```

図20 CDGENにより作成されたLOCAL変数

特定の核種のみを対象とするサブルーチンがある場合、ユーザは、COMMON変数として

```
INCLUDE DIFFUS
```

の他に、

```
INCLUDE DIFFUS#11
```

```
INCLUDE DIFFUS#21
```

```
INCLUDE DIFFUS#31
```

と記述することにより、対象となる変数のアドレスを添字を使わずに、ローカル変数名でアクセスすることができる。

### (3) 初期化用 BLOCK DATA と再計算用ファイル入出力ルーチンの生成

CDGENは最初に、(1)で説明したサイズブロックの生成を終え、ユーザの使用するEOSソースのサイズ、COMDECKを確定する。その後、このファイルを再度入力して、初期化用BLOCK DATAと再計算用ファイル入出力ルーチンを生成する。

BLOCK DATAは2つの例外を除いて全てのCOMMON領域をゼロセットする。例外は次の2つのCOMMON BLOCKである。

#### 例外 (1) COMMON / ZSIZE /

すべてのサイズ変数をBLOCK内に持つCOMMONである。

サイズ変数の値がセットされる。

#### (2) COMMON / ZINDEX /

LOCAL変数を生成する際の付加文字列からインデックスを逆参照するためのテーブル。文字列と番号がセットされる。

前項と同じく最終的なCOMDECKリストを入力して全COMMON変数を再計算用ファイルに出力(入力)するルーチンを生成する。

### 3. ユーザ DATA SET の作成

ユーザが本ライブラリ・システムを用いて、専用のコードを作成する際の作業手順を以下に示す。

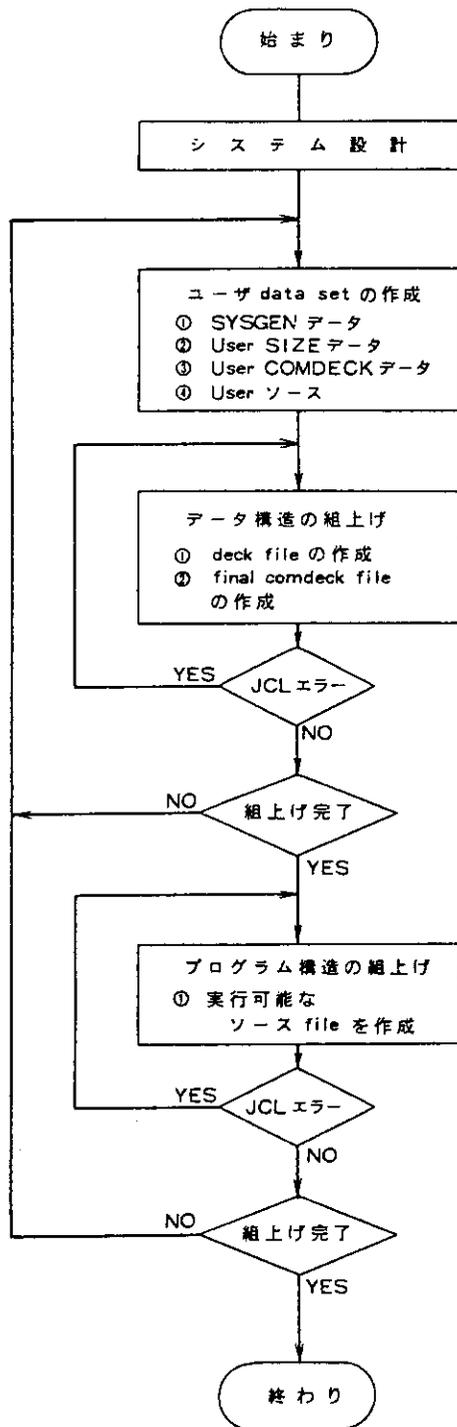


図 2 1 LIBRARY システム組上げ作業の流れ

データ構造の組上げに必要な、#SYSGEN, #SIZE, #COMDECK データ, ユーザソースを作成する方法をここでは記述する。

ライブラリ・システムにおいて、システム組上げ時にユーザが関与するものは、ユーザ・システムファイルである。ユーザ・システムファイルは、次のモジュールから成り立つ。

(1) SYSGEN 記述モジュール

#SYSGEN (ユーザは必ず作成する。)

(2) DATA 記述モジュール

#SIZE (システムの#SIZE以外に追加・変更がなければ作成する必要はない。)

#COMDECK (システムの#COMDECK以外に追加・変更がなければ作成する必要はない。)

(3) PROGRAM 記述モジュール

#DECK (システムのソース以外に追加・変更がなければ作成しない。)

システム組上げ後、ユーザ管理下になるファイルは、上記ユーザ・システムファイルと、組上げの結果作成されたDECKファイル, COMDECKファイル, およびロードモジュールの4種のファイルとなる。

### 3.1 モジュール#SYSGEN

ライブラリ・システムを使って独自のシミュレーションプログラム(または実験解析プログラム: ESC, EAC)を組上る際には、自分のプログラムの機能を表現するオプション・パラメータを入力しなくてはならない。システム記述のパラメータを4段階に分類して考える。

#### 3.1.1 コンパイル時オプションの指定

##### STEP1 プラズマ粒子の選択

対象とするプラズマ中に存在する核種を指定する。ただし、指定されたイオンの密度と温度はすべてがプログラム内部で計算されるか、外部テーブルから与えられるかのいずれかで定義されなければいけない。

表4 プラズマ粒子のオプション・パラメータ

パラメータ名	値	意味
P R	0 1	Proton が存在しない 存在する
D E	0 1	Deutron 同上
T R	0 1	Tritium 同上
H E	0 1	Helium 同上
C A	0 1	Carbon 同上
O X	0 1	Oxygen 同上
F E	0 1	Iron 同上
A R	0 1	Argon 同上
M O	0 1	Molibdenum 同上

STEP 2 基本式を選択

プラズマ中の電子，イオンの温度の式，連続の式，電磁気の式の中でシミュレーションコードの中で解かれる式を指定する。STEP 1 で指定しない粒子（プラズマ中に存在しない粒子）についての式を指定すると誤りとなる。

表 5 基本式のオプション・パラメータ

パラメータ名	値	意味
EQPR	0 1	Protonの連続の式を { 解かない 解く
EQDE	0 1	Deuteron "
EQTR	0 1	Tritium "
EQHE1	0 1	He <sup>+</sup> イオン "
EQHE2	0 1	He <sup>++</sup> イオン "
EQK	0 1	不純物イオン "
EQCZ	0 1	電流の式
EQBP	0 1	ポロイダル磁場の式
EQTEL	0 1	電子温度の式
EQTPR	0 1	Proton温度の式
EQTDE	0 1	Deuteron温度の式
EQTTR	0 1	Tritium温度の式
EQTK	0 1	不純物温度の式
EQTI	0 1	バルクイオン平均温度の式
EQEL	0 1	電子密度の連続の式を { 解く 解かない
EQNHY	0 ~ 2	水素の連続の式の数
EQNTHY	0 ~ 2	水素の温度の式の数

STEP 3 項を選択

STEP 2 で指定された項の中でオプションによって不要となる項の有無を指定する。現在の所NBIおよびFUSION項のみである。

表 6 項のオプション・パラメータ

パラメータ名	値	意味
FUS	0 1	水素イオン2成分の時 { fusion項なし fusion項あり
NBI	0 1 2	NBI項なし 水素1に対してNBI項有り 水素2 "

STEP 4 解法の選択

STEP 3 までで決定された基本式群の中のいくつかを連立させて解くかどうかを指定する。

表7 解法のオプション・パラメータ

パラメータ名	値	意味
CPN	0	水素イオンが2種の時に個別に解く
	1	" 連立させて解く
CPHE	0	Heliumの連続の式を個別に解く
	1	" 連立させて解く
CPT	1	$T_e, T_1$
	2	$T_e - T_1$
	3	$T_e, T_1, T_2$
	4	$T_e, T_1 - T_2$
	5	$T_e - T_1 - T_2$
	6	$T_e, T_i$
	7	$T_e - T_i$

(注) CPTの「意味」の記号

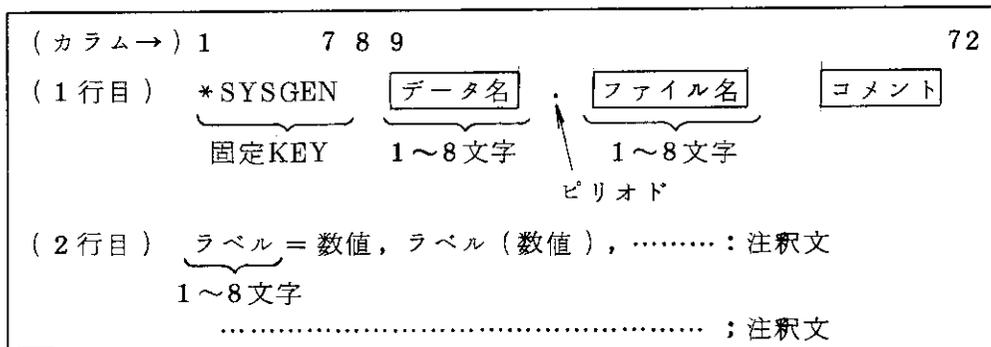
- $T_e$  : 電子温度
- $T_1$  : 水素1の温度
- $T_2$  : 水素2の温度
- $T_i$  : 複数のイオン種についての平均温度
- 「-」 : 連立させて解く
- 「,」 : 個別に解く

例えば、 $T_e, T_1 - T_2$ は電子温度の式は単独で、水素1と水素2の温度は連立させて解くことを意味する。

3.1.2 #SYSGEN記述規約

(1) SYSGENデータの構成

- アスタ(\*)行 (第1行目)
- パラメータ行 (第2行目以降)





```

0100 *SYSGEN #SYSGEN.LIBJT60 === PR,CZ,TE-TPR WITH NBI === (82/08)
0200 PR=0 ; PROTION ION
0300 DE=0 ; DEUTRON ION
0400 TR=0 ; TRITIUM ION
0500 HE=0 ; HELIUM ION
0600 CA=0 ; CARBON ION *** IMPURITY ***
0700 OX=0 ; OXIGEN ION *** IMPURITY ***
0800 FE=0 ; IRON ION *** IMPURITY ***
0900 AR=0 ; ARGON ION *** IMPURITY ***
1000 MO=0 ; MOLIBDENUM *** IMPURITY ***
1100 EQNHY=0 ; EQUATION NUMBER OF HYDROGEN
1200 EQPR=0 ; EQUATION *** PROTON ***
1300 EQDE=0 ; EQUATION *** DEUTRON ***
1400 EQTR=0 ; EQUATION *** TRITIUM ***
1500 EQHE1=0 ; EQUATION *** HELIUM HE+ ***
1600 EQHE2=0 ; EQUATION *** HELIUM HE++ ***
1700 EQK=0 ; EQUATION *** IMPURITY ***
1800 EQNTHY=0 ; ENERGY EQUATION NUMBER
1900 EQTPR=0 ; ENERGY EQUATION *** PROTON ***
2000 EQTDE=0 ; ENERGY EQUATION *** DEUTRON ***
2100 EQTTR=0 ; ENERGY EQUATION *** TRITIUM ***
2200 EQTK=0 ; ENERGY EQUATION *** IMPURITY ***
2300 EQTI=0 ; ENERGY EQUATION *** ION AVERAGE ***
2400 EQTEL=0 ; ENERGY EQUATION *** ELECTRON ***
2500 EQEL=0 ; ELECTRON DENSITIES
2600 EQCZ=0 ; CURRENT DENSITY
2700 EQBP=0 ; MAGNETIC DENSITY
2800 CPN=0 ; SOLUTION TYPE OF HYDROGEN (COUPLE, DECOUPLE)
2900 CPHE=0 ; SOLUTION TYPE OF HELIUM (COUPLE, DECOUPLE)
3000 CPT=0 ; ENERGY SOLUTION TYPE (CPT=1,2,3,4,5,6,7)
3100 ESC=0 ; ESC OPERATION FLAG
3200 FUS=0 ; FUSION FLAG
3300 NBI=0 ; NBI FLAG <<<<<<< END OF DEFAULT VALUE SETTING >>>>>>
3400 PR=1
3500 EQNHY=1
3600 EQPR=1
3700 EQCZ=1
3800 EQNTHY=1
3900 CPT=2
4000 EQTPR=1
4100 EQTEL=1
4200 NBI=1

```

図 2 2 ライブラリ・#SYSGEN データ

作成例として、

- ① バルクイオン：プロトン
- ② 不純物 : なし
- ③ 連続の式 : プロトン密度
- ④ 電流拡散式
- ⑤ エネルギー式 : 電子温度, プロトン温度
- ⑥ NBI

を解く場合のデータが、ISN = 3400 ~ 4200 に示されている。

### 3.2 モジュール#SIZE

組み上げるライブラリ・システムのデータ領域の大きさを指定するデータが、#SIZEの内容である。ユーザは、システム・ファイルの#SIZEで指定されているデータサイズを変更したい時は、またはユーザのソースプログラムで、別のサイズ名を使った場合には、ユーザの#SIZEモジュールで、サイズを定義しなければならない。変更する時は、変更するサイズについてのみ定義すればよい。

このデータは、COMDECK GENERATOR (CDGEN)の入力データとなる。SIZEデータに変更がない場合は、作らなくてよい。

#### 3.2.1 SIZEデータの指定

表8 SIZEデータ

SIZEデータ名	説明	標準値
NVAL	連立して解かれる式の数の最大値	3
MPOINT	外部テーブル点数	10
MFUNC	外部テーブル数	10
MGP	ガス・パフテーブルの数	5
MSH	空間メッシュ数 (主イオン)	30
MZ	空間ゾーン数	5
MENT	中性粒子成分数	5
MEG	中性粒子エネルギースペクトル数	5
MCPNT	プラズマ電流ポイント数	10
MGAU	ガウス積分分点数	20
NCOM	COMMONラベルの数	100
MENBI	NBIのエネルギーレベル数	10
MPNBI	NBIのTIMEテーブルのポイント数	10
NCHARGE	=NK+1 (現在は、システムで自動計算)	
MSHK	空間メッシュ数 (不純物イオン)	30

3.2.2 SIZE 記述規約

(1) SIZEデータの構成

アスター(\*)行 (第1行目)  
 サイズデータ (第2行目以後)

```

    カラム→1 2
          * SIZE ..... アスター行
          MSH=30, NB=5 ; コメント
          AA=1
          BB=2
          CC=3, DD=4
          ⋮
    
```

- (2) アスター行は、必ず第1カラム目から書く。
- (3) SIZE記述は、' \* SIZE 'で始まらねばならない。
- (4) サイズデータは、2~71カラム内に書く。
- (5) サイズデータは、(名前)=(値)の形式で書く。
- (6) 同一行にくり返し書くときは、', 'で区切りを入れる。
- (7) コメントは、';コメント'または':コメント'の形で記述する。コメントは、71カラム以内に納めること。

3.2.3 #SIZEデータの作成例

```

-----*-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-R-----*-----8
NO.6      MODULE NAME #SIZE(1)          BLOCKS      2
          LEVEL      5          DATE      81.12.02  TIME      09.28.51

          ***** APPOINTED MODULE INFORMATION *****

          LEVEL      5          DATE      81.12.02  TIME      12.10.42

*SIZE BULK.SYSTEM(2)          00000100
NVAL=3;MAXIMUM EQUATION NUMBER (COUPLING) 00000200
MPOINT=10          00000300
MFUNC=10          00000400
MGP=5          00000500
MSH=30          00000600
MZ=5          00000700
MENT=5          00000800
MEG=5          00000900
MCPNT=10          00001000
MGAU=20          00001100
NCOM=100          00001200
MENBI=10          00001300
MPNBI=10          00001400
NCHARG=NK + 1          00001500
MSHK=30          00001600
    
```

図 2 3 ユーザ・#SIZE・データ

- (1) モジュール名は'#SIZE'とする。
- (2) 上の例はシステムの'#SIZE'データである。ユーザ・データの場合は'USER'とする。

### 3.3 モジュール#COMDECK

モジュール#COMDECKは、ユーザのコモン・デック・データ (COMDECK DATA) である。このデータは、システムの#COMDECKとともに、COMDECK GENERATOR (CDGEN) の入力となる。

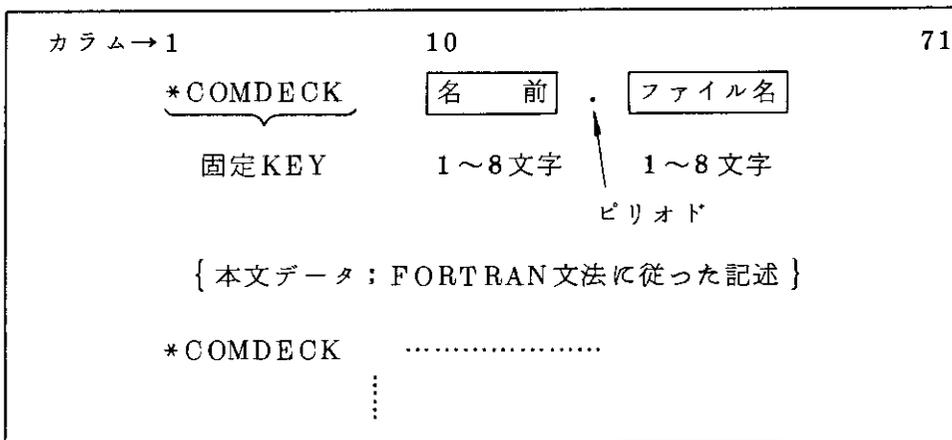
COMDECK DATAは、FORTRAN 文法で許されるものであれば何でもよい。例えば、1つのサブルーチン・プログラム全体であってもよい。ソースプログラム内に、「INCLUDE 名前」のように入れておけば、名前に対応したコモンがその位置にそう入される。

このデータは、ユーザが必要なければ作成しなくてもよい。一方、システムのCOMDECKを変更したい時には、USER COMDECKを作成し、そこに変更するものを登録すればよい。

#### 3.3.1 COMDECKの記述規約

##### (1) COMDECKデータの構成

アスター(\*)行 (第1行目)  
 本文 (第2行目以後)



(2) アスター行は、必ず第1カラム目から書き、第71カラム以内で納めること。

(3) COMDECK記述は、「\*COMDECK」で始まらねばならない。

(4) アスター行の規約

- ① 先頭8文字「\*COMDECK」であること。
- ② 10~71カラム以内に、名前、ファイル名、オプションを記述する。
- ③ 名前、ファイル名、オプションの順に記述する。
- ④ 名前は、1~8文字とする。
- ⑤ ファイル名は、1~8文字で「SYSTEM」以外のものとする。  
原則として、「USER1~3」を用いる。
- ⑥ (名前).(ファイル名)のように、「.」を間に入れる。
- ⑦ 名前の前には、1個以上の を入れること。

名前は、ソースプログラムのINCLUDE名と一致すること。一致するCOMDECKデータがない場合は、エラーとなる。

- (5) 本文データは、FORTRAN 文法に従って記述する。  
つまり、コメント「C」は、第1カラムに、実行文は、第7カラムから記述する。
- (6) 配列の大きさは、#SIZEで指定したサイズ名を使用することができる。
- (7) コモン・ブロックは、アスター行（\*COMDECK ……）で、1つの区切り文と認識され1ブロックとされる。  
“同じ名前”のブロックが存在する場合には、最初のものが有効とされる。

3.3.2 #COMDECK データの作成例

```

*COMDECK          DOUBLE.SYSTEM          00001600
  IMPLICIT REAL*8 (A-H,O-Z)              00001700
*COMDECK          ZSYS.SYSTEM            00001800
C          ----- SYSTEM PARAMETER ----- 00001900
  COMMON/ ZSYS / NAMED(2), NAMEF(2), NTITLE(20) 00002000
*COMDECK          ZSIZE.SYSTEM           00002100
C          ----- PARAMETER SIZE ----- 00002200
  COMMON/ ZSIZE / NION, NB, NK, NHY, NHE,      00002300
  1          NCA, NOX, NAR, NFE,              00002400
  2          NEUT, NBO, NKO, NHYO,            00002500
  3          NEQU, NEQB, NEQTB, NEQHY, NEQHE, NEQTHY, 00002600
  4          NCHARG, NVAL, NPOINT, MSH, MSHK,  00002700
  5          MFUNC, MGP, MZ,                  00002800
  6          MCPNT, MEG, MENBI, MENT, MGAU, MPNBI, NCOM 00002900
      .
      .
*COMDECK          YAVRG.SYSTEM            00009600
C          ----- AVERAGE DENSITY, TEMPERATURE ----- 00009700
  COMMON/ YAVRG / AVN(NION), AVT(NION), AVNE, AVTE, AVNI, AVTI 00009800
*COMDECK          XNT.SYSTEM              00009900
C          ----- ION DENSITIES ----- 00010000
  COMMON/ XNT / DO(MSH,NION), DP(MSH,NION), DN(MSH,NION),      00010100
  1          TO(MSH,NION), TP(MSH,NION), TN(MSH,NION),          00010200
  2          TIO(MSH,), TIP(MSH), TIN(MSH)                      00010300
      .
      .
*COMDECK WORK3.SYSTEM                      00050500
  DIMENSION WW7(NCHARG,MSH), WW8(NCHARG,MSH), WW9(NCHARG,MSH) 00050600
*COMDECK WORK4.SYSTEM                      00050700
  INTEGER IWW1(MSH), IWW2(NCHARG)          00050800
*COMDECK          UCASE.SYSTEM             00050900
  NAMELIST/ UCASE / LHEL, LZEFF            00051000
*COMDECK          USPACE.SYSTEM            00051100
  NAMELIST/ USPACE / RT, RW, RP, RS, ELL   00051200
  
```

図 2 4 #COMDECK データ

この例は、SYSTEMの#COMDECKの一部である。

ユーザが作成する時には、アスター(\*)行を

```
*COMDECK XNT. USER
```

のように、作成すればよい。

### 3.4 ユーザ・ソースモジュールの作成

システムのソース・ファイルの内容を変更したい時や、追加したいソース・ファイルがある場合に、ユーザは、ユーザGEMファイルに、修正または追加するソースモジュールを登録しなければならない。

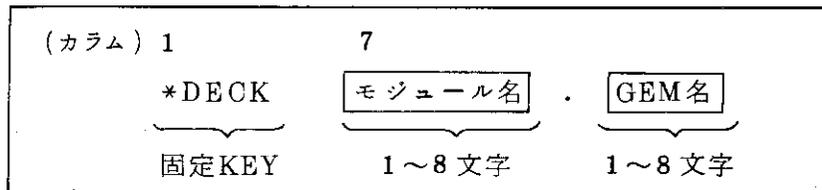
このユーザソースは、ORGを実行することにより、システムソースと置換されたり、追加される。ORGは、モジュール名を認識し、モジュールがGEMモジュールを呼び出している時は、これを呼び出し、ソースを作成する。同一モジュールが出現した時には、これを無視する（最初に現われたものが有効）。行がオプション行であれば、これを評価し、必要に応じてこれを削除する。また、行がプリント行であれば、FORTRAN記述に変換する。ORG実行後、EOSを実行する事により、EOS言語で書かれている部分は、本来のFORTRAN文のみからなるソースに変更され、ユーザが指定したSYSGENデータに従ったソースライブラリが組み上げられる。

ソースプログラム構造記述は、以下より構成される。

- (1) DECK行
- (2) ソース本文（EOS言語も含める）
- (3) オプション行
- (4) 注釈行
- (5) プリント行（ユーティリティ）
- (6) INCLUDE行

#### 3.4.1 ソースモジュール記述規約

- (1) DECK行（モジュール行、アスター行）



- ① ソースモジュールは、\*DECK行で始まらねばならない。
- ② '\*DECK'は、固定のものであり、ORGの認識記号となっている。
- ③ モジュール名、GEM名は7~71カラム以内に記述する。
- ④ GEMファイル上のモジュール名と、DECK行のモジュール名は同一であること。
- ⑤ GEM名は、USER, USER1, ……の如く'SYSTEM'と区別する識別子であり、DD名となる。ORGの出力結果にGEM名が表示されることにより、SYSTEMのソースと区別される。

ユーザは、ソースモジュールを作成する場合はGEM名として、'SYSTEM'以外を使用しなければならない。

```

-----*-----1-----*-----2-----*-----3-----*-----4-----*-----5-----*-----6-----*-----7-R-----*-----8
NO.16  MODULE NAME KAUXV          BLOCKS      3
          LEVEL      8          DATE      81.09.09  TIME      14.47.55

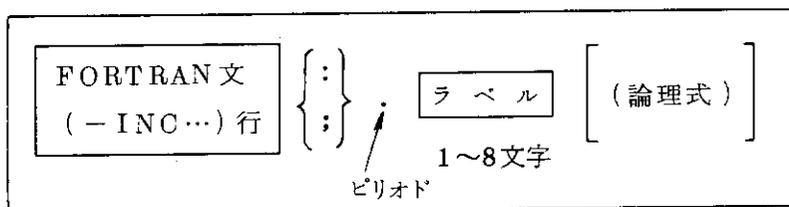
          ***** APPOINTED MODULE INFORMATION *****
          LEVEL      8          DATE      81.10.01  TIME      17.17.16

一致させる
          この例では「SYSTEM」

*DECK KAUXV.SYSTEM          ← DECK 行          00000100
      SUBROUTINE  KAUXV( KTR )          00000200
C*****00000300
C*  KAUXV          AUXILIARY PARAMETERS FOR IMPURITY CALCULATION  *00000400
C*  KAUXV          AUXILIARY PARAMETERS FOR IMPURITY CALCULATION  *00000500
C*                                     *00000600
C*                                     *00000700
C*****00000800
C                                     00000900
      INCLUDE  DOUBLE          00001000
      INTEGER KER          00001100
      INCLUDE ZCONT ; ITER          00001200
  
```

図 2 5 ソースモジュール記述例

(2) オプション行



- ① DECK行以外のどの行でも「:.', ';;.'が出現した時には, オプション行と認識される。
- ② DECK行にオプション行を書込んではいならない。
- ③ ラベルは, 同一モジュール内で有効である。
- ④ あるラベルについては, 最初に出てくる時に論理式部分「(論理式)」を記述し, ラベルに論理値を与える。  
論理値は, 最初のラベルのみ有効である。
- ⑥ ラベルは, 8文字以内とする。
- ⑦ ラベルは, 論理式中で前方参照することが出来る。この場合には, 「&」をラベルの前に付さねばならない。

例     :: A (NA.EQ. 1)  
       :: AN (.NOT.&A)

- ⑧ オプション行は，論理値=TRUE 以外は削除される。
- ⑨ 論理式について：
- a. 論理式は，FORTRANと同様である。  
ただし，算術式は，2項比較演算のみが許される。
  - b. 変数は， $\left\{ \begin{array}{l} \text{SYSGENのオプションパラメータ} \\ \text{既定義ラベル(' \&ラベル'の形で使用)} \end{array} \right.$   
のみ許される。
  - c. 定数は，10進数のみ許される。
  - d. 許される論理演算は  
' . NOT. '，' . OR. '，' . AND. '
  - e. 許される算術比較式は，下のもののみとする。  
' . EQ. '，' . GT. '，' . LT. '，' . GE. '，' . LE. '，' . NE. '，' . NOT. '
- ⑩ ラベルの個数は，1モジュール内で100を越えてはならない。

《注意》

1.  $\left\{ \begin{array}{l} \text{: . A (B. EQ. 1)} \\ \text{... : . A (B. EQ. 1)} \end{array} \right.$  1カラム目  
のように，': .'または': .'から行が始まるのは許されない。
2. : . AB (A + B. GE. C) は許されない。  
' A + B ' は文法エラーとなる。

	*DECK KCONST.USER	00000100
	SUBROUTINE KCONST	00000200
C	-----	I00000300
C	CONST --- PHYSICAL CONSTANTS	I00000400
C		I00000500
C	-----	I00000600
	INCLUDE DOUBLE	00000700
	INCLUDE ZSIZE	00000800
	INCLUDE VPHYS	00000900
	INCLUDE VPHYS#11	00001000
	INCLUDE YCONST	00001100
C		00001200
	DO 10 I=1,NK	00001300
	CMK(I) = CMP=12.DO :.A(CA.EQ.1)	00001400
	CMK(I) = CMP=16.DO :.B(OX.EQ.1)	00001500
	CMK(I) = CMP=40.DO :.C(AR.EQ.1)	00001600
	CMK(I) = CMP=55.DO :.D(FE.EQ.1)	00001700
C		00001800
	MAK(I) = 12 :.A	00001900
	MAK(I) = 16 :.B	00002000
	MAK(I) = 40 :.C	00002100
	MAK(I) = 55 :.D	00002200
C		00002300
	NAK(I) = 1	00002400
	10 CONTINUE	00002500
C		00002600
	RETURN	00002700
	END	00002800

} 使用例

図 2 6 オプション行の使用例

(3) 注釈行

```

/*

```

- ① ' /\* ' で始まる行は注釈行と認識され削除される。
- ② ' /\* ' の前には, '空白' しか置けない。

(4) INCLUDE行

```

8 カラム
- { INC
  INCLUDE } 〔 GEMモジュール名 〕 [ .レベル ] [ /パスワード ]
              [ , SEQ = ] , IN = 〔 DD名 〕

```

- ① GEMモジュール名, DD名は, 必ず指定されねばならない。
- ② ソースプログラム本文の後に記述する。
- ③ モジュール名とGEMモジュール名は同一名にしておく。
- ④ 使用の詳細は, 「GEM 3 使用手引書」<sup>2)</sup>の INCLUDE 制御文の項を参照のこと。

```

CALL KNUTR( DNSAV2, ALPHK,DEB,RKK,RW, RT, TK1HOT, ZMK,      00003700
            NCHARG, NK+1, MSH, NKKP1                          ) 00003800
CALL KTRV01(DNSAV2,NCHARG,1,ZWK1,NKKP1)                      00003900
CALL KSOLTO(ZWK1,KER)                                        00004000
CALL KTRV02(DNSAV2,NCHARG,1,ZWK1,NKKP1)                      00004100
C                                                            00004200
10 CALL KPARAM( KER )                                       00004300
      IF( KER.GT.O)          GO TO 800                      00004400
      GO TO 900                                                    00004500
C                                                            00004600
800 CALL UMSG( 'KAUXV -- KPARAM.' )                          00004700
900 RETURN                                                    00004800
      END                                                        00004900
/* INCLUDE */                                               00005000
- INC KMSSET,IN=SYSTEM                                       00005100
- INC KIONIZ,IN=SYSTEM                                       00005200
- INC KNURT,IN=SYSTEM                                        00005300
- INC KTRV01,IN=SYSTEM                                       00005400
- INC KSOLTO,IN=SYSTEM                                       00005500
- INC KPARAM,IN=SYSTEM                                       00005600
- INC UMSG,IN=SYSTEM                                         00005700

```

図 2 7 注釈行および INCLUDE 行の使用例

## (5) プリント行

.PRINTOPEN, .PAGE, .BLINE, .PRINT, .PRINTCLOSE,  
.CNGFMT の 6 種の機能がある。

詳細は、7.1 PRINT ユーティリティの項を参照のこと。

## (6) サブルーチン名, コモンブロック名の制限

ライブラリ・システムでは, サブルーチン名とコモンブロック名について, その先頭に特定の一文字を割当てることによって, 各々の名前と機能・内容を対応づけている。ユーザは, ユーザ・プログラムを作成する場合, システムで使用される文字以外から始まる名前をサブルーチン名, およびコモンブロック名として用いなければならない。ユーザがシステムで使用する文字を用いた場合, ライブラリ・システムから組上げられるコードの健全性は, 保障されない。システムで使用されている文字を以下に示す。

## ○ サブルーチン名

- K : 不純物
- N : 中性粒子
- U : ユーティリティ
- V : 補助変数計算
- W : 入出力
- X : 下位の制御ルーチン
- Y : 上位の制御ルーチン
- Z : 予備

## ○ コモンブロック名

- K : 不純物
- N : 中性粒子
- U : ネイムリスト
- V : 補助変数
- W : 外部テーブル
- X : 基本変数
- Y : 主プラズマの物性値
- Z : 制御用

## 4. データ構造の組上げ

ユーザの望むプログラムをライブラリ・システムより作成する場合、ユーザは、まずプログラム内容に即したデータ構造（サイズ、コモン変数等）を組上げなければならない。ここでは、データ構造を組上げる方法について説明する。

ライブラリ・システムのデータ構造は、DECKファイルおよび最終COMDECKファイルからなる。DECKファイル（順編成ファイル：PS）は、コモブロックの初期化用ブロックデータおよび再計算用ファイルの入出力ルーチン（「VRIN」，「VROUT」のサブルーチン名で作成される。）である。最終COMDECKファイル（PS）は、システムの提供するサイズ変数およびコモン変数とユーザの指定する変数とからなっている。

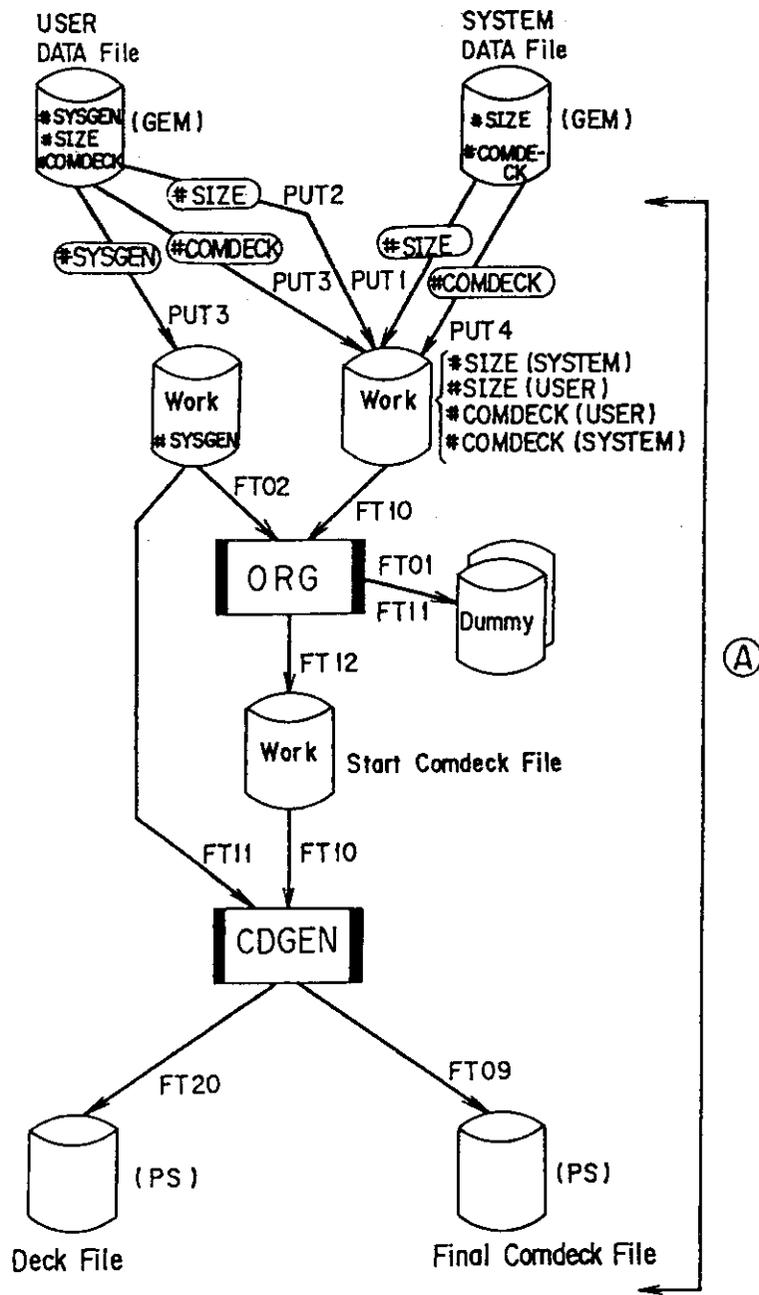
データ構造組上げの概略図を図28に示す。ユーザは、ユーザ・システムファイルに#SYSGENを作成し、また変更があれば、#SIZE、#COMDECKおよびプログラム記述を作成した後、ORG、CDGENを実行することによりDECKファイルおよびCOMDECKファイルが作成され、データ構造の組上げを完了できる。図28に示される一連の処理を行う上で必要なJCLは、TSSコマンド「LIBJCL」を用いて自動的に生成される。

### 4.1 データ構造組上げ用「JCL」の作成

ライブラリ・システムでは、ユーザがJCL作成に時間的労費をしないよう、私用（ライブラリ専用）のJCLマクロおよびTSSマクロコマンドを用意している。ユーザは、それらのマクロコマンドをユーザセッションのTSSコマンドファイルにコピーして置けば、TSSコマンド「LIBJCL」により、一連のJCLを作成することができる。「LIBJCL」の使い方の詳細は6.1 JCLの作成を参照のこと。

例として、CDGEN処理をするためのJCLを、TSSマクロコマンドより作成する。

```
LIBJCL CDGEN  USO('J9999. ABC. FORT')
              COMDECK('J9999. COMDECK. FORT')
              DECK('J9999. DECK. FORT')
```



※ ①の部分の一連の処理のJCLは、TSSコマンドLIBJCLにより自動的に生成される。

図 2 8 データ構造組上げ図

TSS マクロコマンド「LIBJCL」の後のパラメータは、入力オプションである。入力オプションとしては、以下のものがある。

表9 LIBJCLの入力オプション

KEY	CDGEN …… 固定
PASS	JOBのユーザパスワード
T, C, I, W, OPN, NLP, NOTIFY, MSGLVL	} JOBのパラメータオプション
USO	#SYSGEN, #SIZE, #COMDECKモジュールの入っているユーザのGEMファイル名。(full name)
COMDECK	作成される最終コムデック・ファイル名。(full name)
DECK	作成されるデック・ファイル名。(full name)
UNIT	コムデック, デック・ファイルのユニット名。

このコマンドによって作成されたJCLを図29に示す。入力オプションについては、マクロコマンドのデフォルト値として設定することができるので、ユーザはシステムより、TSSマクロコマンドをユーザファイルにコピーした時点で、デフォルト値を書換えておくと、マクロコマンドによるJCL作成は、簡単かつ誤りの少ないものになる。

```

0100 //JCLG JOB
0200 // EXEC JCLG
0300 //SYSIN DD DATA,DLM='++'
0400 // JUSER 71973051,TO.HIRAYAMA,0012.100
0500 T.1 W.2 C.3 P.0 I.4 OPN
0600 OPTP PASSWORD=██████████,MSGLEVEL=(1,1),NOTIFY=J3051
0700 //*****
0800 //***** CONSULTING TEL. 5509 *****
0900 //*****
1000 ++
1100 // DD DSN=J2855.LIBJCL.CNTL(CDGEN),DISP=SHR
1200 // DD DATA,DLM='$$'
1300 //*****
1400 //*****
1500 //*****
1600 /** CDGEN **
1700 //*****
1800 //EXEC CDGEN,LM='J2855.LIBJT60.LOAD',
1900 // USO='J3051.LIB1D.FORT',
2000 // COMDECK='J3051.COMDECK.FORT',DECK='J3051.DECK.FORT',
2100 // SYSTEM='J2855.LIBJT60.FORT',SYSOUT='*',UNIT=TDS,
2200 // PARAM='DBGL=1,INCL=20,GEMN=200,EXPS=1'
2300 //GEMA.SYSIN DD *
2400 - PUT @INC,OUT=SYSUT2,NOID
2500 - PUT #SIZE,OUT=SYSUT2,NOINCEXP,AD
2600 - END
2700 /*
2800 //GEMB.SYSIN DD *
2900 - PUT #SYSGEN,OUT=SYSGEN,NOID
3000 - PUT #SIZE,OUT=SYSUT2,NOINCEXP,NOID,AD
3100 - PUT #COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
3200 - END
3300 /*
3400 //GEMC.SYSIN DD *
3500 - PUT #COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
3600 - END
3700 /*
3800 $$
3900 //

```

図 29 LIBJCL によるデータ組上げ用 JCL (CDGEN) の作成

## 4.2 データ構造組上げの実行例

JCL CDGEN による以下の実行例は，ユーザ GEM ファイルに #SYSGEN データのみ存在する場合である。出力順にリストを記載する。

- ① SYSTEM ファイルからモジュール「@INC」を put.

```
STEP: 1 PUT OPERATION

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
- PUT @INC,OUT=SYSUT2,NOID
JQH255I-I MODULE '@INC' HAS BEEN PUT.

HIGHEST SEVERITY CODE=00
```

図 3 0 - 1 データ構造組上げ実行例

- ② SYSTEM ファイルからモジュール「#SIZE」を put.

```
STEP: 2 PUT OPERATION

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
- PUT #SIZE,OUT=SYSUT2,NOINCEXP,AD
JQH255I-I MODULE '#SIZE' HAS BEEN PUT.

HIGHEST SEVERITY CODE=00
- END

STATISTICS: HIGHEST SEVERITY CODE=00
```

図 3 0 - 2 データ構造組上げ実行例 ( 続き )

- ③ USERファイルからモジュール「#SYSGEN」をput.

STEP: 1 PUT OPERATION

```

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
- PUT #SYSGEN,OUT=SYSGEN,NOID
JQH255I-I MODULE '#SYSGEN' HAS BEEN PUT.

```

HIGHEST SEVERITY CODE=00

図 3 0 - 3 データ構造組上げ実行例 ( 続き )

- ④ USERファイルからモジュール「#SIZE」をput.(この例ではモジュール#SIZEはない)

STEP: 2 PUT OPERATION

```

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
- PUT #SIZE,OUT=SYSUT2,NOINCEXP,NOID,AD
JQH118I-E MODULE #SIZE IS NOT FOUND ON LIBRARY.

```

HIGHEST SEVERITY CODE=08

図 3 0 - 4 データ構造組上げ実行例 ( 続き )

- ⑤ USERファイルからモジュール「#COMDECK」をput.(この例ではモジュールは存在しない)

STEP: 3 PUT OPERATION

```

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
- PUT #COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
JQH118I-E MODULE #COMDECK IS NOT FOUND ON LIBRARY.

```

HIGHEST SEVERITY CODE=08

- END

STATISTICS: HIGHEST SEVERITY CODE=08

図 3 0 - 5 データ構造組上げ実行例 ( 続き )

⑥ SYSTEM ファイルからモジュール「#COMDECK」を put.

STEP: 1 PUT OPERATION

```

*** NEED,SUPPRESS,DEFAULT ***
-   DEFAULT=(NOIDENT)
*****
-   PUT #COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
    JQH255I-I MODULE '#COMDECK' HAS BEEN PUT.
    
```

```

HIGHEST SEVERITY CODE=00
- END
    
```

STATISTICS: HIGHEST SEVERITY CODE=00

図 3 0 - 6 データ構造組上げ実行例 ( 続き )

⑦ ORG 実行結果出力 ( その 1 )

	「*」行の名前 ↓	ファイル識別名 ( 「*」行のファイル名 ) ↓ ( ここでは SYSTEM ファイルを意味 )
** OPTLBL ** (BULK	SIZE	SYSTEM )
** OPTLBL ** (DOUBLE	COMDECK	SYSTEM )
** OPTLBL ** (DBLZZZ	COMDECK	SYSTEM )
** OPTLBL ** (ZSYS	COMDECK	SYSTEM )
** OPTLBL ** (ZCPU	COMDECK	SYSTEM )
** OPTLBL ** (ZSIZE	COMDECK	SYSTEM )
}		
** OPTLBL ** (UNBI	COMDECK	SYSTEM )
** OPTLBL ** (UNBITM	COMDECK	SYSTEM )
** OPTLBL ** (UNEUT	COMDECK	SYSTEM )
1 EQK	0 ←	EQK というオプション名が使用され、結果は「偽」と判定された。
** PTTBLT ** (UNEUT	COMDECK	SYSTEM )

図 3 0 - 7 データ構造組上げ実行例 ( 続き )

⑧ ORG 実行結果出力 ( その 2 )

```
<< SYSTEM GENERATE >>
*SYSGEN #SYSGEN LIBID
=== HPR, JZ, TPR-TE === (82/6)
```

} #SYSGEN データの「\*」行中の名前,  
ファイル名, コメントを出力したもの。

```
*MODULE LABEL
MODULE          GEM
1 BULK          SIZE      SYSTEM
2 DOUBLE        COMDECK   SYSTEM
3 DBLZZZ        COMDECK   SYSTEM
4 ZSYS          COMDECK   SYSTEM
5 ZCPU          COMDECK   SYSTEM
6 ZSIZE         COMDECK   SYSTEM
7 ZCASE         COMDECK   SYSTEM
8 ZIONO         COMDECK   SYSTEM
9 ZSPACE        COMDECK   SYSTEM
```

}

```
80 UIONO        COMDECK   SYSTEM
81 UTIME        COMDECK   SYSTEM
82 UOUT         COMDECK   SYSTEM
83 UCONT        COMDECK   SYSTEM
84 UFORM        COMDECK   SYSTEM
85 UCUR         COMDECK   SYSTEM
86 UGPUF        COMDECK   SYSTEM
87 UDBG         COMDECK   SYSTEM
88 UNBI         COMDECK   SYSTEM
89 UNBITM       COMDECK   SYSTEM
90 UNEUT        COMDECK   SYSTEM
```

⊕ Start COMDECK に put とされたブロック (またはモジュール) を一覧表にまとめたものである。

図 3 0 - 8 データ構造組上げ実行例 ( 続き )

⑨ CDGEN 実行結果出力

\*\*\*\*\* INPUT SYSGEN DATA \*\*\*\*\*

NAME =#SYSGEN  
 FILENAME=LIB1D  
 TITLE = === HPR, JZ, TPR-TE === (82/6)  
 NUMBER OF PARAMETERS( 31 )

#SYSGEMデータの  
 \*SYSGEN行で指定した  
 モジュール名, ファイル  
 名, コメントの内容が出力  
 される。

NO.	NAME	= VAL
1	PR	= 1
2	DE	= 0
3	TR	= 0
4	JE	= 0
5	CA	= 0
6	OX	= 0
7	FE	= 0
8	AR	= 0
9	MO	= 0
10	EQNHY	= 1
11	EQPR	= 1
12	EQDE	= 0
13	EQTR	= 0
14	EQHE1	= 0
15	EQHE2	= 0
16	EQK	= 0
17	EQNTHY	= 1
18	EQTTPR	= 1
19	EQTDE	= 0
20	EQTTR	= 0
21	EQTK	= 0
22	EQTI	= 0
23	EQTEL	= 1
24	EQEL	= 0
25	EQCZ	= 1
26	EQBP	= 0
27	FUS	= 0
28	NBI	= 0
29	CPN	= 0
30	CPHE	= 0
31	CPT	= 2

CDGEN で処理された SYSGEN パラメータ  
 値の結果の出力。

===== INPUT CHECK END == NO ERROR DETECTED.

===== CDGEN END == C.C.(0) ===== DATE 82-10-20

完了コード; ≠ 0 の時には, 誤りを修正して必ず今一度  
 CDGEN 処理をすること。

図 30-9 データ構造組上げ実行例 (続き)

## 5. プログラム構造の組上げ

4章までの作業により，プログラム構造の組上げに必要な準備は完了したことになる。ユーザの作成したソースまたは修正したソースは，ORG，EOSを実行した後，SYSTEMのソースに追加，置換され，コンパイル→リンクの順で，実行可能なロードモジュールが作成される。あるいは，即，実行する段階まで進むこともできる。

ここでユーザは，次の事を注意すべきである。システムのソースを含め，ユーザが作成したソースプログラムは，FORTRAN言語のみのプログラムでなくなっている。つまり，「FORTRAN仕様+EOS言語+その他ライブラリ用仕様（オプション機能，プリント機能などが追加されている）」となっているプログラムであることである。そのためORG処理，EOS処理の2つの前段処理を必ず通さねばならない。これらの処理の後，初めてFORTRANプログラムになる。

処理の手順の概略図を図31に示す。

### 5.1 EOS実行パラメータ

EOS言語で書かれたプログラムを，FORTRANプログラムに変換するための，前処理システムである。EOSを実行するに当たっての実行パラメータ，およびその意味を以下に示す。

実行時のパラメータ

```
// EXEC PGM = EOS22, PARM = 'SLIST = 0, LINECM = 1'
```

実行時のパラメータは，ライブラリシステムでは，SLIST，LINECMの2種を考えればよい。入力方法は，PARM = '……' の形式で必要に応じて指定する。

表10 EOSのキー・パラメータ

パラメータ	意味
SLIST	EOS処理後のソースの出力オプション = 0 ; 出力しない ≠ 0 ; 出力する
LINECM	ソース中の行コメントオプション = 1 ; 行コメントと見なし，その部分を空白としてEOS処理する ≠ 1 ; EOS処理後もコメントはそのまま残す

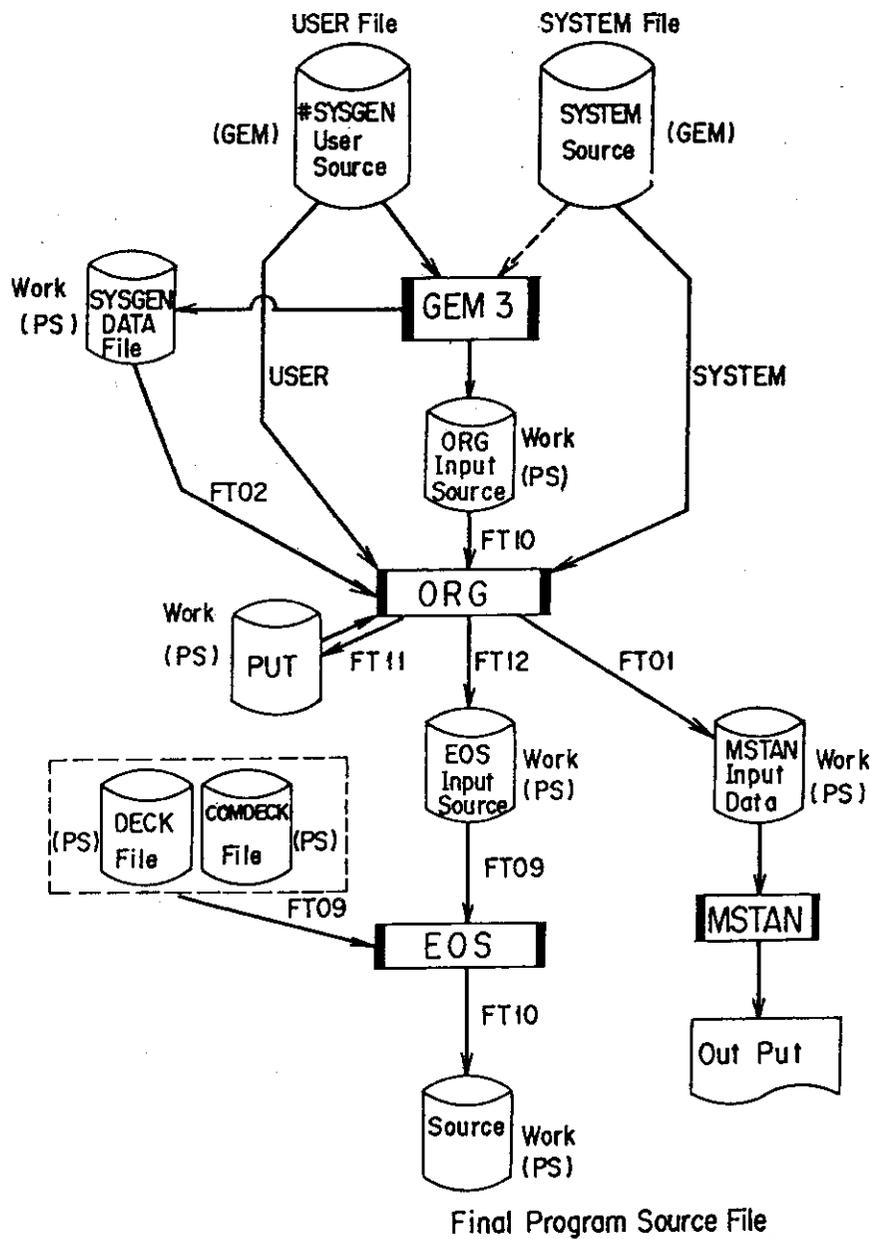


図 3 1 プログラム構造組上げ概略図

## 5.2 ORG実行パラメータ

#SYSGEN データ，およびソースライブラリ中に記述されたプログラム構造記述（3.4 ユーザ・ソースモジュールの作成）に従って，ユーザ・ソースプログラムを組み上げるものが，ツール「ORG」である。

ORG処理をするためには，ユーザGEMファイルとシステムGEMファイルの中より，所定のモジュールを選定し，作業ファイルに「PUT」（GEM PUT 制御文）する必要がある（図31）。この作業ファイルが，ORGの入力ソース・データセットとなる。ここで，

'PUT'したいモジュールの入っているGEMファイルを、「エントリGEMデータセット」と呼び、'PUT'されるモジュールを「エントリ・モジュール」と言う。エントリGEMには、ユーザGEMとシステムGEMの2種類がある。エントリGEMとして、システムGEMを用いる時は、ユーザはライブラリ・システムが用意しているモジュールをそのまま'PUT'すればよい。ユーザGEMをエントリGEMとする場合、ユーザはそのGEM中のINCLUDE展開の有無を、PUTコマンドの中に記述しなければならない。すなわち、ユーザ・エントリ・モジュールが、下位のルーチンを'CALL'している場合、その下位ルーチンをエントリ・モジュール内で'CALL'する(展開する)か否かを、INCLUDE展開の有無によって指示する。下位ルーチンもユーザ・エントリ・モジュールである場合は、必ずINCLUDE展開を指定する。以下に、ORGを起動する際のパラメータ、およびINCLUDE展開の指定方法を説明する。

(1) 実行時のパラメータ

```

//_EXEC _PGM=ORG, PARM='DBGL=2, INCL=20, GEMN=200,
EXPS=1'
    
```

実行時パラメータは、DBGL, INCL, GEMN, EXPSの4種であり必要があれば、PARM='.....'の形式で指定する。

表11 ORGの実行パラメータ

パラメータ	意味
DBGL	デバッグ出力レベル = 0 ; 出力しない 1 ; 下位モジュール呼び出しの出力 2 ; 1およびオプションラベルの出力
INCL	INCLUDE展開レベルの指定 = n ( $\geq 0$ の整数 ) ; nを越えるレベルの INCLUDE展開を行な わない。
GEMN	GEMの起動回数 = n ( 正の回数 ) ; nを越えてはGEM 3を起 動しない。従って INCLUDE展開をしない。
EXPS	INCLUDE展開を行うか否か。 = 0 ; INCLUDE展開をおこなわない。 1 ; " " おこなう。

(2) INCLUDE 展開を動的に変更する場合

ユーザGEMをエントリGEMとする場合、ORG起動中にEXPSパラメータを変更する必要が出てくる。(1)の実行時パラメータは、起動前にしか指定できないので、ORGは起動中に変更を許す機能を持っている。

ORGでは、入力ソース中に、次の行が出現するたびに'EXPS'を変更する。

第1カラム		
* INC	.....	EXPS=1とする
* NOINC	.....	EXPS=0とする

入力パラメータが、例えばEXPS=1であっても、もしモジュール「@NOINC」が途中で'PUT'されれば、その後は「@INC」が現われるまでEXPS=0として処理される。

LIBRARYシステムでは

(GEMモジュール名)	(内容)
@INC	*INC
@NOINC	*NOINC

なるデータが、SYSTEM GEM ファイルに用意されているので、ユーザは必要に応じてこれらをORGの入力ファイル作成時にPUTすることによって、INCLUDE展開モード、または非展開モードを任意に指定できる。

非展開モード

- PUT @NOINC, OUT=DD
- PUT MODULE名, OUT=DD, NOINCEXP
- PUT " , "

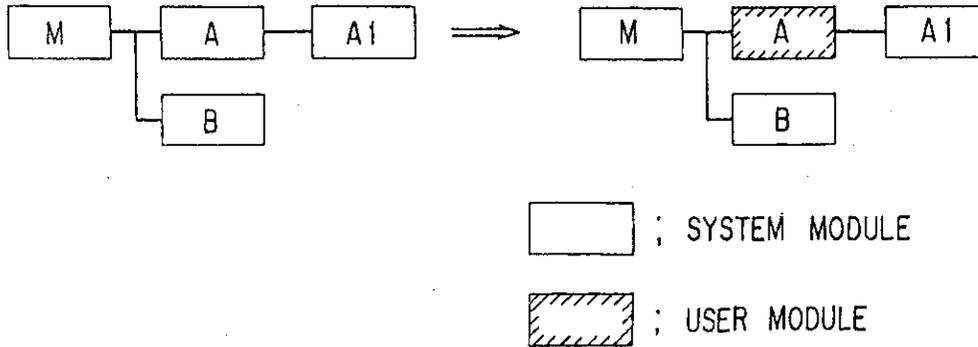
展開モード

- PUT @INC, OUT=DD
- PUT MODULE名, OUT=DD
- PUT " , "

非展開モードでは、PUTパラメータに'NOINCEXP'を必ず指定しなければならない。

エントリGEMがユーザGEMである場合、ユーザは展開モードと非展開モードを明確に区別しなければならない。非展開モード'NOINCEXP'で指定されたユーザ・エントリ・モジュールは、最優先で作業ファイルにPUTされる。従って、ユーザがシステムファイルと置換したい場合、そのユーザ・エントリ・モジュールを、最初に非展開モードでPUTすればよい。以下にその例を示す。

例 1.



① ENTRY GEM; USER GEM

- PUT @NOINC, OUT = DD  
 - PUT A, OUT = DD, NOINCEXP

② ENTRY GEM; SYSTEM GEM

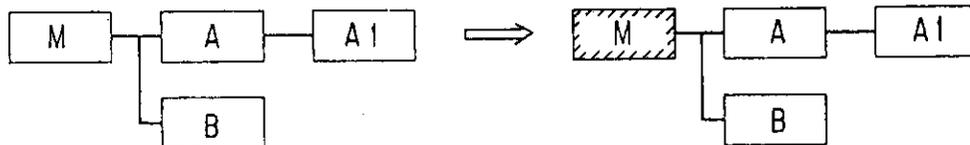
- PUT @INC, OUT = DD  
 - PUT M, OUT = DD

《注》

①, ②の順にPUTする

システム・エントリ・モジュールAを、ユーザ・エントリ・モジュールAと置換する。ただし、Aの下位ルーチンA1は、システムのものを使用するため、ユーザ・エントリ・モジュールAは非展開モードとなる。

例 2.



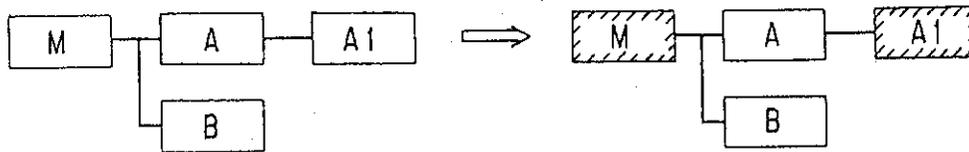
① ENTRY GEM; USER GEM

- PUT @INC, OUT = DD  
 - PUT M, OUT = DD

メインルーチンMを、システム・エントリ・モジュールからユーザ・エントリ・モジュールに置換する。

メインルーチンは、必ず展開モードで指定されなければならない。

## 例3.



## ① ENTRY GEM; USER GEM

```

-PUT @NOINC, OUT = DD
-PUT A1, OUT = DD, NOINCEXP
-PUT @INC, OUT = DD
-PUT M, OUT = DD

```

メインルーチンMと下位ルーチンA1を、システム・エントリ・モジュールからユーザ・エントリモジュールに置換する。A1は、非展開モードでPUTされているが、この例ではA1は最下位のルーチンである。すなわち、その下位に他のルーチンを含んでいないため、展開モードを指定してもよい。一般的に、ユーザ・エントリ・モジュールを用いる場合は、その下位のモジュールから順次、非展開モードでPUTすれば、システム・モジュールと置換され、ユーザの目的とするコードが組上げられることになる。

## 5.3 プログラム構造組上げの実行例

プログラム構造を組上げる際に必要な各種JCL(EOS, ORG, コンパイル, およびリンクなど)は、TSS マクロコマンド 'LIBJCL' によって提供される。ユーザは、端末から

```
LIBJCL OEFL (その他のパラメータ)
```

と入力し、画面上に表示される指示に従って、パラメータを入力して行けば、所定のJCLを得ることができる。LIBJCLで作成されるJCLは、ロードモジュールの領域が確保されていることを前提としているので、ユーザは組上げ用JCLを 'SUBMIT' する前に、ロードモジュール用ファイルを作成しておかなければならない。

ロードモジュールを更進する場合はJCLの外に、TSS マクロコマンド 'LB' を用いて端末上でも処理することができる。コマンド 'LIBJCL' および 'LB' の使用方法については、6.JCL, TSS マクロコマンド 参照のこと。

システム・エントリ・モジュールから、ロードモジュールを新しく作成した場合の簡単な出力例を次に示す。

(1) ユーザファイル中の#SYSGENデータの'PUT'

FACOM OSIV/F4 GEM V02L32 DATE 82.10.27 TIME 16.27.36

-----\*-----1-----\*-----2-----\*-----3-----\*-----4-----\*-----5-----\*-----6-----\*-----7

STEP: 1 PUT OPERATION

\*\*\* NEED,SUPPRESS,DEFAULT \*\*\*  
 - DEFAULT=(NOIDENT)  
 \*\*\*\*\*  
 - PUT #SYSGEN,OUT=SYSGEN,NOID  
 JQH255I-I MODULE '#SYSGEN' HAS BEEN PUT.

HIGHEST SEVERITY CODE=00  
 - END  
 STATISTICS: HIGHEST SEVERITY CODE=00

(2) 展開モードの指定

FACOM OSIV/F4 GEM V02L32 DATE 82.10.27 TIME 16.27.47

-----\*-----1-----\*-----2-----\*-----3-----\*-----4-----\*-----5-----\*-----6-----\*-----7

STEP: 1 PUT OPERATION

\*\*\* NEED,SUPPRESS,DEFAULT \*\*\*  
 - DEFAULT=(NOIDENT)  
 \*\*\*\*\*  
 - PUT @INC,OUT=SYSUT2,NOID  
 JQH255I-I MODULE '@INC' HAS BEEN PUT.

HIGHEST SEVERITY CODE=00  
 - END  
 STATISTICS: HIGHEST SEVERITY CODE=00

(3) メインルーチン「YMAIN」の'PUT'

FACOM OSIV/F4 GEM V02L32 DATE 82.10.27 TIME 16.27.51

-----\*-----1-----\*-----2-----\*-----3-----\*-----4-----\*-----5-----\*-----6-----\*-----7

STEP: 1 GROUP OPERATION  
 - GROUP GG,MODNAME=(YMAIN)  
 JQH023I-I THE GROUP 'GG' WAS DEFINED.

HIGHEST SEVERITY CODE=00

STEP: 2 PUT OPERATION

\*\*\* NEED,SUPPRESS,DEFAULTL \*\*\*  
 - DEFAULT=(NOIDENT)  
 \*\*\*\*\*  
 - PUT GG,OUT=SYSUT2,NOINCEXP,NOID,AD  
 JQH255I-I MODULE 'YMAIN' HAS BEEN PUT.

HIGHEST SEVERITY CODE=00  
 - END

STATISTICS: HIGHEST SEVERITY CODE=00

(4) ユーザエントリモジュールの'PUT' (ここではPUTされていない)

FACOM OSIV/F4 GEM V02L32 DATE 82.10.27 TIME 16.27.57

-----\*-----1-----\*-----2-----\*-----3-----\*-----4-----\*-----5-----\*-----6-----\*-----7

STEP: 1 GROUP OPERATION  
 - GROUP GG  
 JQH023I-I THE GROUP 'GG' WAS DEFINED.

HIGHEST SEVERITY CODE=00  
 - END

STATISTICS: HIGHEST SEVERITY CODE=00

(5) ORG実行結果出力 (モジュールレベル一覧表の出力)

```
<< SYSTEM GENERATE >>
*SYSGEN #SYSGEN LIBID
=== HPR, JZ, TPR-TE === (82/6)
```

```
*MODULE LABEL
MODULE GEM
1 YMAIN DECK SYSTEM
2 YBASE DECK SYSTEM
3 YTRANG DECK SYSTEM
4 YEPILG DECK SYSTEM
5 UPAGE DECK SYSTEM
6 UMSG DECK SYSTEM
7 XDFALT DECK SYSTEM
8 XINPRT DECK SYSTEM
9 VTITLE DECK SYSTEM
10 XINP1 DECK SYSTEM
11 XINP2 DECK SYSTEM
12 XINIT DECK SYSTEM
13 XICAL DECK SYSTEM
    ))
144 NSGVII DECK SYSTEM
145 NYGNR DECK SYSTEM
146 NYCHCK DECK SYSTEM
147 NGICEF DECK SYSTEM
148 VTEIE DECK SYSTEM
149 VTEII DECK SYSTEM
150 VTPFII DECK SYSTEM
151 VTGDII DECK SYSTEM
152 NGPLS DECK SYSTEM
153 NGPUF DECK SYSTEM
154 NGINIT DECK SYSTEM
```

(6) EOS 処理の出力

完了コード (C.C) が 0 の場合には, 誤りを直して必ず最初からランしなおすこと。

===== EOS END ===== C.C.( 0) ERR.NUM.( 0) CARD.( 12062)

(7) リンケージ出力メッセージ

FACOM OSIV/F4 LINKAGE EDITOR V03L16 V03L16 DATE 82.10.27 TIME 16.35.29

OPTIONS SPECIFIED - NOMAP,LIST,LET  
GENERATED - NOXREF

\*\*\*VALUES IN EFFECT\*\*\* SIZE< DEFAULT USED >=(251904,83968),LINECOUNT=60  
MAX. LENGTH OF OUTPUT TEXT BLOCK = 12288

JQA0000 \*  
JQA0000 ENTRY MAIN  
JQA0000 NAME TEMPNAME

\*\* MEMBER NAME\*\* TEMPNAME NOW ADDED TO LIBRARY.

\*\*TTR\*\*( 00 / 07 - 92 / 05 ) \*\*AUTHORIZATION CODE\*\*( 0 )  
\*\*NOW 97 TRACKS(S) LEFT UNUSED IN LIBRARY COVERING 1 EXTENT(S). < RLSE >

## 6. JCL, TSS マクロコマンド

ライブラリ・システムでは、ライブラリ専用のJCLマクロ、およびTSS マクロコマンドを用意しており、ユーザのJCL作成およびロードモジュールの更新を容易にしている。

ライブラリ・システムには、専用マクロコマンドを使用するために、以下のファイルが登録されている。

1° TSS コマンド : J2855.LIBTSS.CLIST

2° JCL マクロ : J2855.LIBJCL.CNTL

LIBTSS.CLISTには、JCL用マクロコマンドを起動する'LIBJCL' マクロコマンドと、端末上で、ロードモジュールを更新するための'LB'マクロコマンドがはいっている。LIBJCL.CNTLには、5種類の私用JCLマクロコマンドCDGEN, ORG, EOS, MSTAN, STANが登録されている。ユーザはマクロコマンド'LIBJCL'を打鍵し、必要なパラメータを入力するだけで、上の5種類のJCLを手にすることができる。ユーザが、システムの提供するマクロコマンドを利用するためには、「J2855.LIBTSS.CLIST」内のモジュール「LB」, 「LIBJCL」をユーザセッションのTSSコマンド・ファイルに、前もってコピーしておかなければならない。また、コピーしたモジュール内のデフォルト値(ユーザID, パスワードなど)を、ユーザの使用に適した値に直しておく必要がある。

### 6.1 JCLの作成

#### 6.1.1 LIBJCLマクロコマンド

ユーザが、「J2855.LIBTSS.CNTL」より、コピーしたモジュール「LIBJCL」は、図32-1となっている。ユーザは、実線で示された箇所のパラメータのデフォルト値を、ユーザ専用の値に直して登録しておく(図32-2)。その他のパラメータの説明を表12に示す。

```

PROC 1 KEY                               /* KEYWORD                */-
TSSMAC('TSSMAC.CLIST')                  /* TSS MACRO OF USER     */-
PASSW('12345678,TU.LIBRARY,0012.100')    /* PASSWORD               */-
PASSW(T)                                  /* PASSWORD               */-
T(1) C(3) I(4) W(2) P(0) OPN(OPN)        /* JOB PARAM.            */-
NLP(' ') NOTIFY(JXXXX) MSGLVL('(0.1)')   /*                          */-
SYSOUT(*)                                 /* SYSOUT                 */-
DBGL(1) INCL(20) GEMN(200) EXPS(1)       /* ORG PARAMETER         */-
USO('JXXXX.LIBJT60.FORT')                /* USER GEM FILE         */-
USO1('JXXXX.LIBJT60.FORT')               /*                          */-
USO2('JXXXX.LIBJT60.FORT')               /*                          */-
DDN1(USER) DDN2(USER2) DDN3(USER3)       /* USER DDNAME           */-
SLIST(0) LINECM(1)                       /* EOS PARAMETER         */-
COMDECK('JXXXX.COMDECK.FORT')           /* FINAL COMDECK FILE   */-
DECK('JXXXX.DECK.FORT')                  /* DECK FILE             */-
UNIT(TDS)                                 /* COMDECK,DECK :: UNIT */-
SPARAM('1 2 5') SLEV(12)                 /* STAN INPUT DATA     */-
NAME(NONAME) ELVL(W)                     /* FORTGE OPTION        */-
A('NOSOURCE,NOISN,NOMAP')               /*                          */-
B(PRINT) CALL(BYNAME) TRBK(GOSTMT)       /*                          */-
STATIS(NOSTATIS) OBJS('30,10')          /*                          */-
LIBLOAD('JXXXX.LIBLOAD') GL('.LOAD')     /* LIBRARY LOAD MODULE */-
LMN(TEMPNAME)                             /* LOAD MODULE NAME     */-
ENTRY(MAIN)                               /* ENTRY NAME           */-
AL(NOMAP) BL(LIST)                       /* LINKAGE PARAMETER    */-
PRVLIB('SYS9.NO')                        /*                          */-
DATA('JXXXX.RUNDATA.DATA(XXXX)')        /* RUN DATA            */-
WRKS('100,50')                            /* CONDENSE PARAMETER   */-
CSPC('100,50,10') CUNIT(TSSWK)          /*                          */-
AR('ERRCUT=0')                            /* RUN PARAMETER        */-
/*                                          */-
/***** DON'T UPDATE FOLLOWING DATA *****/-
/*****                                     *****/-
/*****                                     *****/-
UPS(30) CNL(NO)                            /* LINKAGE PARAMETER    */-
GGS('SYS9.GGS')                           /*                          */-
GOSYSIN('DDNAME=SYSIN')                   /* SYSIN DDNAME         */-
NEW                                          /* CREATE LOAD MODULE   */-
/*
/*****
/*****
/***** CONSULTING TEL. (5509) *****/-
/*****
/*****

```

図 3 2 - 1 TSS コマンド 'LIBJCL' : システム

```

0100 PROC 1 KEY /* KEYWORD */-
0200 TSSMAC('TSSMAC.CLIST') /* TSS MACRO OF USER */-
0300 PASS('██████████████████████████████████████') /* PASSWORD */-
0400 PASSW(██████████) /* PASSWORD */-
0500 T(1) C(3) I(4) W(2) P(0) OPN(OPN) /* JOB PARAM. */-
0600 NLP(' ') NOTIFY(J3051) MSGLVL('(1,1)')/* */-
0700 SYSOUT(*) /* SYSOUT */-
0800 DBGL(1) INCL(20) GEMN(200) EXPS(1) /* OGR PARAMETER */-
0900 USO('J3051.LIB1D.FORT') /* USER GEM FILE */-
1000 USO1('J3051.LIB1D.FORT') /* */-
1100 USO2('J3051.LIB1D.FORT') /* */-
1200 DDN1(USER1) DDN2(USER2) DDN3(USER3) /* USER DDNAME */-
1300 SLIST(0) LINECM(1) /* EOS PARAMETER */-
1400 COMDECK('J3051.COMDECK.FORT') /* FINAL COMDECK FILE */-
1500 DECK('J3051.DECK.FORT') /* DECK FILE */-
1600 UNIT(TDS) /* COMDECK,DECK :: UNIT */-
1700 SPARAM('1 2 5') SLEV(12) /* STAN INPUT DATA */-
1800 NAME(NONAME) ELVL(W) /* FORTGE OPTION */-
1900 A('NOSOURCE,NOISN,NOMAP') /* */-
2000 B(PRINT) CALL(BYNAME) TRBK(GOSTMT) /* */-
2100 STATIS(NOSTATIS) OBJS('30,10') /* */-
2200 LIBLOAD('J3051.LIB1D') QL('.LOAD') /* LIBRARY LOAD MODULE */-
2300 LMN(TEMPNAME) /* LOAD MODULE NAME */-
2400 ENTRY(MAIN) /* ENTRY NAME */-
2500 AL(NOMAP) BL(LIST) /* LINKAGE PARAMETER */-
2600 PRVLIB('SYS9.NO') /* */-
2700 DATA('J3051.LIB1D.DATA(TEST1)') /* RUN DATA */-
2800 WRKS('100,50') /* CONDENSE PARAMETER */-
2900 CSPC('100,50,10') CUNIT(TSSWK) /* */-
3000 AR('ERRCUT=0') /* RUN PARAMETER */-
3100 /* */-
3200 /***** */-
3300 /***** DON'T UPDATE FOLLOWING DATA *****/-
3400 /***** */-
3500 /* */-
3600 UPS(30) CNTL(NO) /* LINKAGE PARAMETER */-
3700 GGS('SYS9.GGS') /* */-
3800 GOSYSIN('DDNAME=SYSIN') /* SYSIN DDNAME */-
3900 NEW /* CREATE LOAD MODULE */-
4000 /*

```

図 3 2 - 2 TSS コマンド 'LIBJCL' ユーザ

表12-1 TSS マクロコマンドLIBJCLのパラメータ

パラメータ	意味	デフォルト値	備考
KEY	作成したいJCLの形成を指定するもの。	なし	○位置パラメータである。 O: ORG E: EOS F: コンパイル L: リンク R: 実行 S: STAN M: MSTAN
PASS	使用者, 課室等の情報		
PASSW	パスワード		
TSSMAC	ユーザのTSSコマンドの入っているファイル名。	'TSSMAC.CLIST'	ユーザ識別子はつけない。
T C I W P OPN NLP NOTIFY MSGLVL		1 3 4 2 0 OPN  '(0, 1)'	
SYSOUT	SYSOUT 指定	*	
DBGL INCL GEMN EXPS	「ORG」の実行時の入力パラメータ	1 20 200 1	
USO	ユーザの { #SIZE #SYSGEN #COMDECK } とユーザのソースが入っている。	'J2855.LIBJT60.FORT'	USOのファイルには #SIZE, #SYSGEN が存在するので必ずユーザのファイル名を指定しなければならない(完全修飾詞付名)。
USO1	ユーザのソースファイル	'J2855.LIBJT60.FORT'	USOで指定したファイル以外にソースの入っているファイルがなければ, デフォルト値またはUSOと同じファイル名にする。
USO2	ユーザのソースファイル		
DDN1	USOのファイルに対するDD名	USER	○ユーザのソースおよび #COMDECK 中に <u>-INC□, IN=ABC</u> なる INCLUDEの指定があるときの「ABC」がこのDD名に対応する ○DDN2, DDN3についても同じ
DDN2	USO1のファイルに対するDD名	USER2	
DDN3	USO3のファイルに対するDD名	USER3	

パラメータ	意 味	デフォルト値	備 考
LINECM SLIST	「EOS」実行時の入力 パラメータ	0 1	
COMDECK	COMDECKファイル名	なし	○完全修飾詞付名で指定する。 ○「CDGEN」をRUNする ときには、この指定名で新 規にファイルが作成される。
DECK	DECKファイル名	なし	
UNIT	COMDECK, DECK 両 ファイルを作成するとき のUNIT名	TDS	両ファイルとも5トラックの 領域で充分である。
SPARAM SLEV	「STAN」実行時の入力 パラメータ	'1_2_5' 12	
NAME ELVL  A  B CALL TRBK STATIS OBJS	FORTGEのオプション	NONAME W 'NOSOURCE, NOISN, NOMAP' PRINT BYNAME GOSTMT NOSTATIS '30, 10'	.....オブジェクトの領域
LIBLOAD  LOAD	組み上げシステムのロー ドモジュール ロードモジュールの内容 識別子	なし  ' .LOAD '	内容識別子なし
LMN ENTRY	ロードモジュール名 エントリ名	TEMPNAME MAIN	
AL BL	「LINK」実行時入力パ ラメータ	NOMAP LIST	
WRKS  CSPC PRVLIB CUNIT	「LINK」時のワークエ リア CONDENSE 時のワー クエリア PRIVATE LIBRARY CONDENSE 用ワークフ ァイルのUNIT	'100, 50'  '100, 50, 10' 'SYS9.NO' TSSWK	
DATA AR	実行時の入力データファ イル 実行時の入力パラメータ	なし  'ERRCUT=0'	○完全修飾詞付名

表12-2 TSS マクロコマンドLIBJCLのパラメータ

パラメータ	意味	デフォルト値	備考
UPS CNTL GGS	「LINK」時のパラメータ	30 NO 'SYS9.GGS'	
GOSYSIN	SYSINのDD名	'DDNAME=SYSIN'	

以上のパラメータについては、システムのデフォルト値がセットされているのでユーザはかたがへに変更しないこと。

表12-3 TSS マクロコマンドLIBJCLのパラメータ

その他のパラメータ (値なしキーワードパラメータ)

パラメータ	意味	デフォルト値	備考
NEW	○ロードモジュールを初めて作成するとき指定する	指定なし	○ロードモジュールの新規作成のとき NEW と指定する。

マクロコマンド 'LIBJCL' を用いて、JCLを作成する場合、どのような内容のJCLを作成するかを指定するキー・パラメータを入力しなければならない。キー・パラメータとしては、以下のものがある。

表13 LIBJCLのキーパラメータ

DCGEN	:	CDGEN用JCL作成の指示
O	:	ORG "
E	:	EOS "
F	:	FORTGE "
L	:	LKEDUP "
R	:	LMGO "
M	:	MSTAN "
S	:	STAN "

ここで、パラメータ 'CDGEN' は、単独で指定されなければならない。'CDGEN' 以外は、合せて指定することができる。組合せ例には以下の物がある。

表 1 4 LIBJCL キーパラメータ組合せ例

KEY	ORG	EOS	MSTAN	STAN	FORTGE	LKEDUP	LMGO
OEFL	○	○			○	○	
OEFLR	○	○			○	○	○
OM	○		○				
OES	○	○		○			
R							○
OEMS	○	○	○	○			

キー・パラメータの O, E, M, S, F, L, R の並び順は任意である。

6.1.2 LIBJCL の使用例

マクロコマンド 'LIBJCL' の使用例を図 3 3 に示す。ここでは、実行をおこなうための JCL を作成している。

```

LIBJCL R ← KEY パラメータ (その他, パラメータ入力があるときには, その値を入力すること)
<<< T(1) W(2) C(3) P(0) I(4) OPN >>> LIBJCL の起動
<<< OPTP PASSWORD=MSGLEVEL=(1,1),NOTIFY=J3051 >>> } デフォルト値の出力

ENTER ; ; T W C P I OPN NLP OR 'ENTER-KEY' } デフォルト値を修正する
====>3 4 3 0 4 CLS NGT } 時入力する
ENTER ; ; MSGLVL NOTIFY OR 'ENTER-KEY' } (デフォルト値でよい)
====> } → RET キーを入力

KEY = R ← KEY パラメータ値の出力

IF YOU HOPE RE-ENTER KEYWORD ====> ← KEY パラメータ値を変更するとき
                                         入力する
                                         (変更なし → RET キーを入力)

00100 //JCLG JOB
00200 // EXEC JCLG
00300 //SYSIN DD DATA,DLM='++'
00400 // JUSER 71973051,TO.HIRAYAMA,0012.100
00500 T.3 W.4 C.3 P.0 I.4 CLS NGT
00600 OPTP PASSWORD=MSGLEVEL=(1,1),NOTIFY=J3051
00700 //*****
00800 //***** CONSULTING TEL. 5509 *****
00900 //*****
01000 //RUN EXEC LMGO,PNM=TEMPNAME,LM='J3051.LIB1D',Q='.LOAD',
01100 // SYSOUT='*' } 作成された JCL の内容
01200 //RUN.SYSIN DD DSN=J3051.SYSDATA.DATA(TEST),DISP=SHR
01300 //RUN.FT01F001 DD SYSOUT=*
01400 //*RUN.FT20F001 DD DSN=JXXXX.RESTART,DISP=SHR }
01500 //RUN.FT20F001 DD DUMMY } リスタートファイルの指定
01600 ++
01700 //
END OF DATA SET
    
```

図 3 3 LIBJCL の使用例











```

PROC O TSSMACC('TSSMAC.CLIST')          /* FILE OF TSS MACRO */
COMDECK('JXXXX.COMDECK.FORT')          /* COMDECK FILE OF USER */
DECK('JXXXX.DECK.FORT')                /* DECK FILE OF USER */
USO('JXXXX.LIBJT60.FORT')              /* FILE NAME OF SOURCE */
USO1('JXXXX.LIBJT60.FORT')             /* */
USO2('JXXXX.LIBJT60.FORT')             /* */
USER(USER) USER1(USER1) USER2(USER2)  /* DDNAME OF SOURCE */
ORGPRM('DBGL=2,INC,=20,GEMN=100,EXPS=1') /* ORG PARAMETER */
EOSPRM('LINECM=1,SLIST=0')             /* DOS PARAMETER */
LIBLOAD('JXXXX.LIBLOAD.LOAD')          /* LIBRARY LOAD MODULE */
/*
/*****
/***** FOLLOWING DATA ;;; DON'T UPDATE *****/
/*****
SYSMOD(#SYSGEN)                        /* MODULE NAME ;; SYSGEN */
PRINTF('EOSPRW.LIST')                  /* PRINT FILE */
EOS(ON)                                 /* EOS OPTION */
GO(NOGO) SEQ(NOSEQ) FIX(FIXED)         /* COMPILER OPTION */
DATAFILE('J2855.#SYSDATA.DATA(STAB)') /* CONTENT IF AS FOLLOWS */
/* INCLUDE FLOAD(TEMPNAME) */
/* ENTRY MAIN */
/* NAME TEMPNAME(R) */
NEW                                     /* LOAD MODULE NEW OR NOT */
FILE                                    /* WORK FILE CREATE OPT. */
ID(J2855)                               /* ID NUMBER */
/*****
/***** CONSULTING TEL. (5509) *****/
/*****

```

図 35-1 TSS コマンド 'LB' システム

```

0100 PROC 0 TSSMAC("TSSMAC.CLIST")          /* FILE OF TSS MACRO      */-
0200      COMDECK("J3051.COMDECK.FORT")     /* COMDECK FILE OF USER  */-
0300      DECK("J3051.DECK.FORT")          /* DECK FILE OF USER     */-
0400      USO("J3051.LIBLD.FORT")          /* FILE NAME OF SOURCE   */-
0500      USO1("J3051.LIBLD.FORT")         /*                          */-
0600      USO2("J3051.LIBLD.FORT")         /*                          */-
0700      USER(USER1) USER1(USER2) USER2(USER3) /* DDNAME OR SOURCE     */-
0800      ORGPRM("DBGL=2, INGL=20, GEMN=100, EXPS=1") /* ORG PARAMETER        */-
0900      EOSPRM("LINECM=1, SLIST=0")       /* EOS PARAMETER         */-
1000      LIBLOAD("J3051.LIBLD.LOAD")      /* LIBRARY LOAD MODULE   */-
1100      /*                                  */-
1200      /*****                               *****/-
1300      /*****      FOLLOWING DATA ;;; DON'T UPDATE      *****/-
1400      /*****                               *****/-
1500      SYSMOD( #SYSGEN )                  /* MODULE NAME ;; SYSGEN */-
1600      PRINTER("LIBPRT.LIST")            /* PRINT FILE           */-
1700      EOS(ON)                            /* EOS OPTION           */-
1800      GO(NOGO) SEQ(NOSEQ) FIX(FIXED)    /* COMPILER OPTION     */-
1900      DATAFILE("J2855.#SYSDATA.DATA(STAB)") /* CONTENT IF AS FOLLOWS */-
2000      /* INCLUDE FLOAD(TEMPNAME)        */-
2100      /* ENTRY MAIN                      */-
2200      /* NAME TEMPERATURE(R)            */-
2300      NEW                                /* LOAD MODULE NEW OR NOT */-
2400      FILE                                /* WORK FILE CREATE OPT. */-
2500      ID(J2855)                          /* ID NUMBER            */-
2600      /*****                               *****/-
2700      /*****      CONSULTTING TEL. (5509)      *****/-
2800      /*****                               *****/-

```

図 35-2 TSS コマンド 'LB' : ユーザ

表 15-1 TSS マクロコマンド LB のパラメータ

パラメータ	意味	デフォルト値	備考
TSSMAC	ユーザの TSS コマンドの入っているファイル名	'TSSMAC.CLIST'	ユーザ識別子はつけない
COMDECK	COMDECK ファイル名	なし	○ファイル名は完全修飾詞付名で入力 ○「CDGEN」により作成されたファイルのファイル名
DECK	DECK ファイル名		
USO	#SIZE ユーザの #SYSGEN #COMDECK とユーザのソースが入っているファイル名	'J2855.LIBJT60.FORT'	USO には #SIZE, #SYSGEN が存在するので必ずユーザのファイル名を指定しなければならない。 (完全修飾詞名)
USO1	ユーザのソースファイル名	'J2855.LIBJT60.FORT'	USO で指定したファイル以外にソースの入っているファイルがなければ、デフォルト値または USO と同じファイル名にする。
USO2	ユーザのソースファイル名		

パラメータ	意味	デフォルト値	備考
USER	USOのファイルに対するDD名	USER	<ul style="list-style-type: none"> <li>○ユーザソースおよび#COMDECK中に -INC □, IN=ABC なるINCLUDE指定があるときの「ABC」がこのDD名に対応する。</li> <li>○USER1, USER2についても同じ。</li> </ul>
USER1	USO1のファイルに対するDD名	USER1	
USER2	USO2のファイルに対するDD名	USER2	
ORGPRM	「ORG」実行時の入力パラメータ	'DBGL=2, INCL=20, GEMN=100, EXPS=1'	
EOSPRM	「EOS」実行時の入力パラメータ	'LINECM=1, SLIST=0'	
LIBLOAD	組み上げシステムのロードモジュールファイル名	なし	<ul style="list-style-type: none"> <li>○ファイル名は完全修飾詞付名。</li> <li>○領域が確保されていること。</li> </ul>

⊛ 以上のパラメータのデフォルト値については、使用者が任意にセットする。

表15-2 TSSマクロコマンドLBのパラメータ

パラメータ	意味	デフォルト値	備考
SYSMOD	SYSGEN データの入っているモジュール名	#SYSGEN	
EOS	EOSに対する実行オプション	ON	「ORG」処理を一度実行した後、再度LBを起動したいが、「ORG」とばして直接EOSから実行したいときEOS(OFF)と指定すればよい。
GOSEQFIX	コンパイラオプション	NOGO NOSEQ FIXED	
DATAFILE	LINKの入力データ用ファイル名	'J2855.#SYSDATA.DATA.(STAB)' データの内容は INCLUDE FLOAD (TEMPNAME) ENTRY MAIN NAME TEMPNAME(R) となっている。	<ul style="list-style-type: none"> <li>○完全修飾詞付名とする。</li> <li>○ユーザのデータファイルを使用したいときは、そのファイル名を指定する。</li> <li>○ロードモジュールのアップデートの時に使用される。</li> </ul>
PRINTF	出力用ファイルのファイル名	'LIBPRT.LIST'	<ul style="list-style-type: none"> <li>○ユーザ識別なしの名</li> <li>○出力用のワークファイルを使用するのみ</li> <li>○このファイルを使用したいときLBのスタート前にALLOCATIONしておけばよい。</li> </ul>

⊛ 以上のパラメータのデフォルト値については、システムのデフォルト値が定められているので、ユーザはかたてに変更しないこと。

表15-3 TSSマクロコマンドLBのパラメータ

その他のパラメータ (値なしキーワードパラメータ)

パラメータ	意味	デフォルト値	備考
NEW	○ロードモジュールを初めて作成する時に指定するもの。	指定なし	○ロードモジュールの新規作成のとき NEW と指定する。
FILE	○ワークファイルを新規作成するか否かを指定するもの (一日ワークファイル)。	指定なし	○一日の初めにLBを実行するときに FILE と指定する。

## 6.2.2 LBの使用例

TSSコマンド'LB'を用いて、システム・ファイルの「XTBSET」, 「UVADD」を、再度リンクし直した例を図36に示す。ここで、ユーザGEMファイルは、「J2855.TE12SYS.FORT」である。

```

LB          <<<<      12/03/81 14:58:33      >>>>
MODULE '#SYSGEN
$$ GEM3 'J2855.TE12SYS.FORT' START.  WHEN UPDATE END, ENTER 'END'
THE GROUP 'A' ' WAS DEFINED.
GEM3
END
$$ ENTER MODULE NAME OR GROUP NAME ==> XTBEST(N)---@NOINCをPUTする
PUT @NOINC,OUT=DD2,NOID,AD
MODULE '@NOINC
' HAS BEEN PUT.
END
GEM3 'J2855.TE12SYS.FORT' OLD
PUT XTBSSET,OUT=DD2,BOID,NOINC,AD
MODULE XTBSSET
IS NOT FOUND ON LIBRARY.
$$ ENTER MODULE NAME OR GROUP NAME ==> (RET)を入力した
END
===== END OF MODULES PUTTING ( FILE : J2855.TE12SYS.FORT ) =====
$$ IF PUT MODULE OF OTHER FILE, ENTER DSN (FULL NAME)
====> 'J2855.LIBJT60.FORT'
$$ GEM3 'J2855.LIBJT60.FORT' START.  WHEN UPDATE END, ENTER 'END'
GEM3 'J2855.LIBJT60.FORT' OLD
GROUP A
THE GROUP 'A' ' WAS DEFINED.
GEM3
END
$$ ENTER MODULE NAME OR GROUP NAME ==> XTBSSET(N) 上で@NOINCがPUTされたので指定しなくともよい
END
GEM3 'J2855.LIBJT60.FORT' OLD
PUT @NOINC,OUT=DD2,NOID,AD
MODULE '@NOINC
' HAS BEEN PUT.
END
GEM3 'J2855.LIBJT60.FORT' OLD
PUT XTBSSET,OUT=DD2,NOID,NOINC,AD
MODULE 'XTBSSET
' HAS BEEN PUT.

```

ユーザのソースをUPDATE  
 したい時には、GEM3の  
 UPDATEモードに入ればよい。

{ . I → @INCをPUTする  
 { . 指定なし → 何もPUTしない

ユーザファイルにMODULE「XTBSSET」  
 が存在しなかったことを示す。

他のソースファイルにENTRYしたい時に入力  
 する。そうでないときには(RET)を入力。

図 3 6 L B の使用例

```

$$ ENTER MODULE NAME OR GROUP NAME ==> UVADD
PUT UVADD,OUT=DD2,NOID,NOINC,AD
MODULE 'UVADD
' HAS BEEN PUT.
$$ ENTER MODULE NAME OR GROUP NAME ==> (RET)入力した
END
===== END OF MODULES PUTTING (FILE : J2855.LIBJT60.FORT) =====

$$ IF PUT MODULE OF OTHER FILE, ENTER DSN (FULL NAME)
(R)入力した
<<< ORG START >>>
CALL 'J2855.LIBJT60.LOAD(ORG)' 'DBGL-2,INCL=20,GEMN=100,EXPS=1'
                                     |
                                     |-----> ORGのSTART

** OPTLBL ** (XTBSET DECK SYSTEM )
1 K 0

** OPTLBL ** (UVADD DECK SYSTEM )

** PTTBLT ** (UVADD DECK SYSTEM )

<< SYSTEM GENERATE >>
*SYSGEN #SYSGEN TEL2SYS
=== DE,PR,BP,TE-T1-T2 === (81/10)

*MODULE LABEL
MODULE GEM
1 XTBSET DECK SYSTEM
2 UVADD DECK SYSTEM
<<< ORG END (CODE=0) >>>

<<< EOS START >>>
CALL 'J2678.FREP.LOAD(EOS22)' 'LINECM=1,SLIST=0'
===== EOS END ===== C.C.( 0) ERR.NUM.( 0) CARD.( 1790)
<<< EOS END (CODE=0) >>>
                                     |
                                     |-----> EOSのSTART

$$ IF COMPILER, ENTER 'C' ==> C ← 'O'を入力すればコンパイルに入る。'O'を入力すればコンパイルに入らない。
$$ COMPILER START

```

図 3 6 (つづき)

```

FORT PSW1.FORT FIXED NOGO NOSEQ NOSOURCE OBJ(PSW3.OBJ) PRINT(*)
$$ COMPILER O.K. (CODE=0)
$$ IF LINK, ENTER 'L' ==> L ← 'L'を入力すればLINKに入る。'O'を入力すればコンパイルに入らない。

CONDENSE 'J2855.LIBTE12.LOAD'
CONDENSE PROCEEDING, BACK UP DATA SET J2855.CONDENSE WAS CREATED
ALLOC DA('J2855.LIBTE12.LOAD') F(FLOAD) SHR
LINK (PSW3.09J 'J2855.#SYSDATA.DATA(STAB)') F LOAD('J2855.LIBTE12.LOAD') PRINT(*)
FACOM OSIV/F4 LINKAGE EDITOR V03L16 DATE 81.12.03 TIME 15.07.41

OPTIONS SPECIFIED - NONE
GENERATED - NOXREF, NOMAP, NOLIST
***VALUES IN EFFECT*** SIZE< DEFAULT USED >=(251904,83968)

MAX. LENGTH OF OUTPUT TEXT BLOCK = 18432

**MEMBER NAME** TEMPNAME NOW REPLACED IN LIBRARY.
**TTR**( 41 / 08 - 83 / 07 ) **AUTHORIZATION CODE**( 0)
**NOW 47 TRACK(S) LEFT UNUSED IN LIBRARY COVERING 4 EXTENT(S).
FREE DA('J2855.LIBTE12.LOAD')
$$ LINKAGE END, NOW J2855.LIBTE12.LOAD UFDATED (CODE=0)
<<<< LB END (CODE=0) >>>>
READY
    
```

LINKAGEのメッセージ

図 36 (つづき)

## 7. ユーティリティ

ライブラリ・システムには、ユーザのシステム組上げ時の負担を軽減するために、ユーティリティが用意されている。ユーティリティには以下のものがある。

- 1) PRINT ユーティリティ
- 2) MSTAN ユーティリティ
- 3) STAN ユーティリティ

PRINT ユーティリティは、タイトル、メッセージおよび各種変数の出力を行う際に、変数の書式 (FORMAT 文) を省略することを可能にする。MSTAN および STAN はプログラム解析をおこない、サブルーチン・トリートリーなどを提供するユーティリティである。

### 7.1 PRINT ユーティリティ

'PRINT' の機能は、出力の際の FORMAT 形式をユーザが作成する必要を省き、簡便なコマンド行を記述するだけで、必要とする FORMAT 形式を作成する点にある。'PRINT' は、ソースプログラム内で、第一カラムが、'.' または '+' で始まる形式で記述され、ORG により FORTRAN 文に置換される。

使用規約は以下の通りである。

- 1) 1 モジュール内で、'.PRINTOPEN', '.PRINTCLOSE' により、'PRINT' ユーティリティの使用、および終了を宣言する。

• PRINTOPEN  $\lfloor$  n<sub>1</sub> [, n<sub>2</sub> [, n<sub>3</sub> [, n<sub>4</sub> ]]]  
 • PRINTCLOSE

n<sub>1</sub> : 出力機番  
 n<sub>2</sub> : 書式  
 n<sub>3</sub> : 頁当りの行数  
 n<sub>4</sub> : 行当りのカラム数

'PRINTOPEN' は、1 エレメント内で1回しか記述できない。

- 2) 宣言文に 'DIMENSION LBLZ(10)' を入れておくこと。
- 3) モジュールの最後に、  
 - INC UFMPRT, IN = SYSTEM  
 を記述する。
- 4) プリント行では、第1カラムが  
 '.' を、コマンド行  
 '+' を、継続行  
 と言う。コマンド行には、以下の種類がある。

① 改頁, 改行

<ul style="list-style-type: none"> <li>• PAGE_(n) 改頁行の改行数</li> <li>• BLINE_n</li> </ul>	n ; 改行数
---	---------

② タイトル, メッセージ

<ul style="list-style-type: none"> <li>• TITLE_ { ' ..... / . ' }           変数名</li> </ul>
--

タイトルの内容をリテラル入力するときは ' / . ' で終了していなければならない。

③ PRINT 行

<ul style="list-style-type: none"> <li>• PRINT_ { 単変数           配列要素           配列名 (I1-I2(, I1'-I2'[, ...])) } , { ' 単位 ' } [ , ... ]</li> </ul>
--

- I1-I2 ; 出力範囲 ( I1 から I2 まで )
- 単変数, 配列要素は, スカラーとして認識される。
- スカラーとベクタの混在は, 出力が保証されない。

④ 書式変更

<ul style="list-style-type: none"> <li>• CNGFMT_n</li> </ul>	n = 0 ; メッシュの R の値 を書く ≠ 0 ; 書かない
--	---

図 37-1 に, PRINT ユーティリティの使用例を示す。このユーティリティは, 「ORG」によって FORTRAN 文に変換される。図 37-2 に, 変換された内容を示す。図 37-3 に, 実際の出力例を示す。

```

INCLUDE ZCONT                                00002100
INCLUDE SCASE                                00002200
DIMENSION LBLZ(10)                           00002300
C                                              00002400
.PRINTOPEN NWT                               00002500
.PAGE                                         00002600
.BLINE 4                                     00002700
.TITLE ' *****'                          00002800
+*****/. '                                00002900
.TITLE ' *****'                          00003000
+*****/. '                                00003100
.TITLE ' *****'                          00003200
+*****/. '                                00003300
.TITLE ' *****          INITIAL CALCULATION 00003400
+*****/. '                                00003500
.TITLE ' *****'                          00003600
+*****/. '                                00003700
.TITLE ' *****'                          00003800
+*****/. '                                00003900
.TITLE ' *****'                          00004000
+*****/. '                                00004100
.BLINE 4                                     00004200
.TITLE ' ===== PHYSICAL CONSTANTS =====/. ' 00004300
.PRINT PAI, ' ',CMUO, ' ',EPSO, ' ',CK, ' ',CME, ' ',CMP, ' ',EK, ' ',CK, ' ' 00004400
.BLINE 4                                     00004500
.TITLE ' ===== MASS, CHARGE STATES, ATOMIC NUMBER ===/. ' 00004600
.PRINT CM(1-NION), ' KG ',NA(1-NION), ' ',MA(1-NION), ' ' 00004700
.PAGE                                         00004800
.BLINE 2                                     00004900
.TITLE ' ===== MESH POINT VALUES =====/. ' 00005000

)

.PRINTCLOSE                                  00007300
C                                              00007400
RETURN                                       00007500
END                                          00007600
/* INCLUDE */                               00007700
- INC KINITL,IN=SYSTEM :.A                 00007800
- INC UFMPRT,IN=SYSTEM                      00007900

```

図 37-1 PRINT ユーティリティ

```

18      INCLUDE ZCONT                                00002200
19      INCLUDE ZCASE                                00002300
20      DIMENSION LBLZ(10)                          00002400
C
21      MUSZ = 1H-
22      6000 FORMAT( A4, A2, I2, A1, I2, IX )
23      CALL UFMOPN( NWT      ,0      ,60      ,137      )
24      CALL UFMOPN( 2      )
25      CALL UFMOPN( 4      )
26      CALL UFMMSG( ' *****
*****/.'
)
27      CALL UFMMSG( ' *****
*****/.'
)
28      CALL UFMMSG( ' *****
*****/.'
)
29      CALL UFMMSG( ' *****          INITIAL CALCULATION
*****/.'
)
30      CALL UFMMSG( ' *****
*****/.'
)
31      CALL UFMMSG( ' *****
*****/.'
)
32      CALL UFMMSG( ' *****
*****/.'
)
33      CALL UFMOPN( 4      )
34      CALL UFMMSG( ' ===== PHYSICAL CONSTANTS =====/.' )
35      CALL UFMOPN( 'PAI      ' , '      ' , LBLZ )
36      CALL UFMOPN( LBLZ(3), PAI      , 0      , 0      )
37      CALL UFMOPN( 'CMUO      ' , '      ' , LBLZ )
38      CALL UFMOPN( LBLZ(3), CMUO      , 0      , 0      )
39      CALL UFMOPN( 'EPSO      ' , '      ' , LBLZ )
40      CALL UFMOPN( LBLZ(3), EPSO      , 0      , 0      )
41      CALL UFMOPN( 'CK      ' , '      ' , LBLZ )
42      CALL UFMOPN( LBLZ(3), CK      , 0      , 0      )
43      CALL UFMOPN( 'CME      ' , '      ' , LBLZ )
44      CALL UFMOPN( LBLZ(3), CME      , 0      , 0      )
45      CALL UFMOPN( 'CMP      ' , '      ' , LBLZ )
46      CALL UFMOPN( LBLZ(3), CMP      , 0      , 0      )
47      CALL UFMOPN( 'EK      ' , '      ' , LBLZ )
48      CALL UFMOPN( LBLZ(3), EK      , 0      , 0      )
49      CALL UFMOPN( 'CK      ' , '      ' , LBLZ )
50      CALL UFMCLR

```

C

図 37-2 PRINT ユーティリティ



## 7.2 MSTANユーティリティ

MSTAN (Module STructure ANalyzer) は、ライブラリ・システムが提供するコード作成のための支援ツールである。機能としては、インクルード・モジュールのツリー情報、および未定義ルーチン名などの診断情報を入力する。

MSTAN を起動するための JCL は、TSS マクロコマンド

```
LIBJCL OM
```

により作成される。MSTAN についての入力パラメータはない。JCL の作成例は、図 34-3 に示されている。MSTAN による診断情報の出力例を、図 38 に示す。

未定義ルーチン、またはモジュール名の違った名前がエントリ文で定義されている場合、それらルーチンを

```
-INC AAA, IN=000
```

と、ソース中にインクルードしたときには、未定義ルーチンとして以下のように処理される。

```
+AAA .
  ↑
  未定義であることを示す
```



### 7.3 STANユーティリティ

STAN (STRUCTURE ANALYZER for FORTRAN program) は、FORTRAN のプログラムの開発、保守および検査などを支援するツールである。これは FORTRAN のソースプログラムを静的に走査して、ルーチンの呼出し関係や、データの相互関連を調べ、プログラムの現状を正確に反映したドキュメントを提供する。

STAN は以下の機能を持っている。

(0) 各種統計および診断情報

- 全カード枚数、コメント数、サブルーチン数、コモンプロック数、未定義ルーチン数などの全体統計。
- 未定義ルーチン表、呼出ルーチン。
- 基本外部関数表、呼出ルーチン。
- 各ルーチン名表、カード枚数、コメント数など。
- コモンプロック名表。
- エントリー・ルーチンの帰属表

上記(0)ブロックは、無条件に出力される。以下(1)～(5)はオプションで、JCL の EXEC 文のパラメータ、OPT (n<sub>1</sub>, n<sub>2</sub>, ...) で、必要な番号集合を n<sub>1</sub>, n<sub>2</sub>, ... に与えることにより選択出力される。

(1) ツリー図：OPT (1) 機能オプション 1

プログラムの階層構造をツリー図に表わす。

JCL の EXEC 文のパラメータでツリー図の編集をおこなう。

(2) ルーチン/コモンプロック相互関係図：OPT (2) 機能オプション 2

各ルーチンと各コモンプロックの相互参照関係を見やすいクロス表にして示す。

(3) ルーチン呼出、被呼出関係表：OPT (3) 機能オプション 3

ツリー図では、ルーチンの被呼出関係がわかりにくい。ここでは、各ルーチンの呼出し、被呼出し関係をシンボリックなリストにして表わす。使用コモンプロック名を示す。

(4) コモンプロック使用ルーチン名リスト：OPT (4) 機能オプション 4

コモンプロックが使用されているルーチンのリストを示す。

(5) 呼出関係の実引数/仮引数の相互チェック：OPT (5) 機能オプション 5

全ルーチンの、実引数と仮引数の数の違いについて、相互チェックをおこなう。

パラメータの指定は、JCL の EXEC 文の PARM オプションの文字列として与える。

//STAN EXEC PGM=STAN, PARM='パラメータ列'

表16 STANの入力パラメータ

パラメータ列	デフォルト
解析対象ファイル名 <sup>1)</sup>	空白
[OPT ( 整数1 整数2 ... ) ] <sup>2)</sup>	OPT ( 1 2 3 4 5 )
[LEV ( 整数 ) ] <sup>3)</sup>	LEV ( 12 )
[CUT ( 整数1 整数2 ) ] <sup>4)</sup>	CUT ( 2 4 )
[SP ( 整数 ) ] <sup>5)</sup>	SP ( 0 )

## 指定例

例1) PARM = 'STAN OPT ( 1 2 ) LEV ( 6 )'

解析対象ファイル名がSTANで機能オプションは1と2, レベルは6とする。

例2) PARM = 'SAP5'

解析対象ファイル名はSAP5で, 他のオプションはすべてデフォルトをとっている。

=注=

## 1) 解析対象ファイル

順データセット ( PS ) 形式の解析対象のファイル名, この名前はSTAN出力リストのヘッドタイトル部に記入するために与えるもので, 8文字以内とする。

## 2) 機能オプション番号の集合 OPT ( n1 ... )

OPT ( 1 2 ) や OPT ( 1 2 4 5 ) などと指定する。この番号は機能オプション番号で1~5をとる。内容はSTANの機能の1~5の番号に対応している。

## 3) 出力カラムレベル LEV ( n )

出力リスト制御で, ツリー図の指定レベル強制カットと全出力カラム巾指定をかねている。 $N \leq 6$  のとき全出力は72カラム以内に制御されTSS ( TSO ) の利用や, A4 へのコピーに便利になっている。

## 4) ツリー図多重参照カット CUT ( n1 n2 )

共通ルーチンの下位レベルのカットで, n1 は主ツリー図のカットすべき参照回数, n2 はカットしたサブツリー図のさらにカットすべき参照回数となっている。n1 = n2 = 0 とするとツリー図は全展開される。

## 5) ツリー図の行間あけ指定 SP ( n )

n ≥ 1 のときツリー図を見やすく1行あけで表示する。

STANを起動するために必要なJCLは,

```
LIBJCL OES
```

により作成される。JCLの作成例は図34-4に示されている。STANによる解析例を以下に示す。

STAN の出力例

STAN ( V1.6 )

VRIN	IS A ENTRY OF	VROUT
UBLINE	IS A ENTRY OF	UPAGE
URCHCK	IS A ENTRY OF	UICHCK
UFMPRT	IS A ENTRY OF	UFMOPN
UFMCLR	IS A ENTRY OF	UFMOPN
UFMCLS	IS A ENTRY OF	UFMOPN
UFMPAG	IS A ENTRY OF	UFMOPN
UFMBLN	IS A ENTRY OF	UFMOPN

STAN ( V1.6 ) (0) SUMMARY INFORMATIONS

## TOTAL ACCOUNTING INFORMATIONS

NO.OF SOURCE CARDS	.....	14739	
NO.OF ISN	.....	8593	
NO.OF COMMENT LINES	.....	3674	
NO.OF MAIN PROGRAMS	.....	1	
NO.OF SUBROUTINS( UNDEF )	..	165	( 2)
NO.OF FUNCTION SUB'S( UNDEF )..		3	( 14)
NO.OF BLOCK DATA SUB'S	.....	4	
NO.OF ENTRY ROUTINS	.....	35	
NO.OF COMMON BLOCKS	.....	125	

## UNDEFINED ROUTINS

1 ABS	11 EXP
2 ALOG	12 FLOAT
3 AMINI	13 IABS
4 ATAN	14 MOD
5 CLOCKM	15 SIGN
6 DABS	16 SQRT
7 DATE	
8 DEXP	
9 DLOG	
10 DSQRT	

STAN ( V1.6 )	(0) ROUTINE NAME	PA
1	EXPINT	11 KMSHCV
2	KCONST	12 KPRSET
3	KIDIST	13 KSAVE
4	KINITL	14 KTRVO1
5	KINPL	15 KTRVO2
6	KINTGC	16 KXSET
7	KINTGS	17 UBLINE
8	KION	18 UCMDMP
9	KMESH	19 UCMPPT
10	KMSCAL	20 UCNGMS
21	UCNVR	21 UCNVR
22	UCNVRG	22 UCNVRG
23	UCOUL	23 UCOUL
24	UDBGSM	24 UDBGSM
25	UDECOP	25 UDECOP
26	UDXDR	26 UDXDR
27	UEQUAT	27 UEQUAT
28	UERMSG	28 UERMSG
29	UFMBLN	29 UFMBLN
30	UFMCLR	30 UFMCLR
31	UFMCLS	31 UFMCLS
32	UFMCNV	32 UFMCNV
33	UFMMSG	33 UFMMSG
34	UFMOPN	34 UFMOPN
35	UFMPAG	35 UFMPAG
36	UFMPRT	36 UFMPRT
37	UFRQMB	37 UFRQMB
38	UFRQNC	38 UFRQNC
39	UICHCK	39 UICHCK
40	UINTGC	40 UINTGC
41	UINTGS	51 UMTS04
42	UMATML	52 UMTS05
43	UMSG	53 UMTS06
61	UPAGE	61 UPAGE
62	URCHCK	62 URCHCK
63	URCHTB	63 URCHTB
71	UTGRD	71 UTGRD
72	UVADD	72 UVADD
73	UVCRSV	73 UVCRSV

STAN (V1.6) (1) TREE -- MAIN STRUCTURE

..LEVEL-0...LEVEL-1...LEVEL-2...LEVEL-3...LEVEL-4...

@MAIN --+-YBASE --+-XDFALT

```

I      +-VRIN
I      +-USAME
I      +-VTITLE--+-DATE
I      I      +-UPAGE
I      I      --UBLINE
I      +-XINP2 --+-UICHCK
I      I      +-URCHCK
I      I      +-UERMSG
I      I      +-UFMOPN-
I      I      +-UFMPAG
I      I      +-UFMMSG
I      I      +-UFMBLN
I      I      +-UFMCNV
I      I      +-UFMPRT
I      I      +-UFMCLR
I      I      --UFMCLS
I      +-KION. -
I      +-XINP1 --+-UICHCK
I      I      +-URCHCK
I      I      +-UFMOPN-
I      I      +-UFMPAG
    
```

← 未定義ルーチン又は、外部関数であることを示す。

← SUB TREE があることを示す (下のレベルに子ルーチンがあることを示す。)



STAN ( V1.6 ) (5) ARGUMENT NUMBER CHECK

ARGUMENT NUMBER AND SUBPROGRAM TYPE CHECK

NO MISMATCH WAS DETECTED

STAN == NOMAL END ==

## 8. システム使用例

本章では、LIBJT60の使い方を具体的な例で示す。ユーザが、初めてライブラリ・システムを使う場合は項目(1)から、またユーザ・エントリ・GEMを作成しシステムの一部を置換する場合は、項目(2)から実行する。

## (1) TSSマクロコマンドのコピー

ユーザは、システムが提供するTSSマクロコマンドを登録するためのPOファイルを、「TSSMAC.CLIST」名で確保する。次に、「J2858.LIBTSS.CLIST」内のモジュール「LB」、および「LIBJCL」を、同じモジュール名で「TSSMAC.CLIST」にコピーする。「LB」は、端末上でロードモジュールを更新するためのマクロコマンドであり、またLIBJCLは各種JCLを作成するためのマクロコマンドである。ユーザは、これらマクロコマンド内のデフォルト値を、ユーザ専用の値に変更して登録しておかなければならない。図39-1, 2に、TSSマクロコマンド「LB」、および「LIBJCL」の内容を示す。実線

```

0100 PROC 0 TSSMAC("TSSMAC.CLIST")                /* FILE OF TSS MACRO */-
0200 COMDECK("J3051.COMDECK.FORT")                /* COMDECK FILE OF USER */-
0300 DECK("J3051.DECK.FORT")                      /* DECK FILE OF USER */-
0400 USO("J3051.LIB1D.FORT")                       /* FILE NAME OF SOURCE */-
0500 USO1("J3051.LIB1D.FORT")                      /* */-
0600 USO2("J3051.LIB1D.FORT")                      /* */-
0700 USER(USER1) USER1(USER2) USER2(USER3) /* DDNAME OR SOURCE */-
0800 ORGPRM("DBGL=2,INGL=20,GEMN=100,EXPS=1") /* ORG PARAMETER */-
0900 EOSPRM("LINECM=1,SLIST=0")                   /* EOS PARAMETER */-
1000 LIBLOAD("J3051.LIB1D.LOAD")                   /* LIBRARY LOAD MODULE */-
1100 /* */-
1200 /*****
1300 /***** FOLLOWING DATA ;;; DON'T UPDATE *****/-
1400 /*****
1500 SYSMOD(#SYSGEN)                                /* MODULE NAME ;; SYSGEN */-
1600 PRINTER("LIBPRT.LIST")                         /* PRINT FILE */-
1700 EOS(ON)                                         /* EOS OPTION */-
1800 GO(NOGO) SEQ(NOSEQ) FIX(FIXED)                /* COMPILER OPTION */-
1900 DATAFILE("J2855.#SYSDATA.DATA(STAB)") /* CONTENT IF AS FOLLOWS */-
2000 /* INCLUDE FLOAD(TEMPNAME) */-
2100 /* ENTRY MAIN */-
2200 /* NAME TEMPERATURE(R) */-
2300 NEW                                             /* LOAD MODULE NEW OR NOT */-
2400 FILE                                            /* WORK FILE CREATE OPT. */-
2500 ID(J2855)                                       /* ID NUMBER */
2600 /*****
2700 /***** CONSULTING TEL. (5509) *****/-
2800 /*****

```

図39-1 TSSコマンド「LB」

```

0100 PROC 1 KEY /* KEYWORD */-
0200 TSSMAC('TSSMAC.CLIST') /* TSS MACRO OF USER */-
0300 PASS(' ████████████████████████████████████████ ') /* PASSWORD */-
0400 PASSW(██████████) /* PASSWORD */-
0500 T(1) C(3) I(4) W(2) P(0) OPN(OPN) /* JOB PARAM. */-
0600 NLP(' ') NOTIFY(J3051) MSGLVL('(1,1)')/* */-
0700 SYSOUT(*) /* SYSOUT */-
0800 DBGL(1) INCL(20) GEMN(200) EXPS(1) /* OGR PARAMETER */-
0900 USO('J3051.LIB1D.FORT') /* USER GEM FILE */-
1000 USO1('J3051.LIB1D.FORT') /* */-
1100 USO2('J3051.LIB1D.FORT') /* */-
1200 DDN1(USER1) DDN2(USER2) DDN3(USER3) /* USER DDNAME */-
1300 SLIST(0) LINECM(1) /* EOS PARAMETER */-
1400 COMDECK('J3051.COMDECK.FORT') /* FINAL COMDECK FILE */-
1500 DECK('J3051.DECK.FORT') /* DECK FILE */-
1600 UNIT(TDS) /* COMDECK,DECK :: UNIT */-
1700 SPARAM('1 2 5') SLEV(12) /* STAN INPUT DATA */-
1800 NAME(NONAME) ELVL(W) /* FORTGE OPTION */-
1900 A('NOSOURCE,NOISN,NOMAP') /* */-
2000 B(PRINT) CALL(BYNAME) TRBK(GOSTMT) /* */-
2100 STATIS(NOSTATIS) OBJS('30,10') /* */-
2200 LIBLOAD('J3051.LIB1D') QL('.LOAD') /* LIBRARY LOAD MODULE */-
2300 LMN(TEMPNAME) /* LOAD MODULE NAME */-
2400 ENTRY(MAIN) /* ENTRY NAME */-
2500 AL(NOMAP) BL(LIST) /* LINKAGE PARAMETER */-
2600 PRVLIB('SYS9.NO') /* */-
2700 DATA('J3051.LIB1D.DATA(TEST1)') /* RUN DATA */-
2800 WRKS('100,50') /* CONDENSE PARAMETER */-
2900 CSPC('100,50,10') CUNIT(TSSWK) /* */-
3000 AR('ERRCUT=0') /* RUN PARAMETER */-
3100 /* */-
3200 /****** */-
3300 /****** DON'T UPDATE FOLLOWING DATA ***** */-
3400 /****** */-
3500 /* */-
3600 UPS(30) CNTL(NO) /* LINKAGE PARAMETER */-
3700 GGS('SYS9.GGS') /* */-
3800 GOSYSIN('DDNAME=SYSIN') /* SYSIN DDNAME */-
3900 NEW /* CREATE LOAD MODULE */-
4000 /*

```

図 39-2 TSS コマンド 'LIBJCL'

部が、ユーザ専用の値に置き換えられた箇所である。ただし、ユーザ GEM ファイル名を「LIB1D.FORT」、ロードモジュール名を「LIB1D.LOAD」、また実行のためのデータファイル名を「LIB1D.DATA」としている。

(2) ユーザ・ファイルの確保

ユーザは、システムを使用する前に 2 本のファイルを確認して置かなければならない。1 本は、エントリ用のユーザ GEM ファイルで、他はそのロードモジュールである。ここでは、

```

ユーザ GEM ファイル : LIB1D.FORT
ロードモジュール   : LIB1D.LOAD

```



定義の処理が従来と異なるため、使用しない他のパラメータは全て0セットしておかなければならない。

#### (4) DATA 記述

ユーザは、目的とするコードにおいてシステムと異なる配列の大きさ、またはコモン変数名などを使用する場合、ユーザエントリGEM(LIB1D.FORT)に「#SIZE」、および「#COMDECK」名のモジュールを登録し、変更内容を記述する。記述規約については、「3.2モジュール#SIZE」、および「3.3モジュール#COMDECK」を参照のこと。ここでは、中性粒子計算で用いられるテーブル用のサイズ変数(図41-1)とコモン変数(図41-2)を新たに追加している。

```
*SIZE #SIZE.USER1      00000100
MTBL1 = MSH*48          00000200
MTBL2 = (MSH+MGAU)*MSH*18 00000300
MTBL3 = MSH*MSH*6+MSH*10 00000400
```

図41-1 #SIZE

```
*COMDECK ZTABLE.USER1      00000100
COMMON /TABLE1/           00000200
> FDNO (MSH,MSH,2) , FUDI (MSH,MSH,2) , FDNX (MSH,MSH,2) , 00000300
> FFLO (MSH,MSH,2) , FFLI (MSH,MSH,2) , 00000400
> FENO (MSH,MSH,2) , FENI (MSH,MSH,2) , FCTO (MSH,MSH,2) , 00000500
> FDNOK(MGAU,MSH,2) , FDNIK(MGAU,MSH,2) , FDNXK(MGAU,MSH,2) , 00000600
> FELOK(MGAU,MSH,2) , FELIK(MGAU,MSH,2) , 00000700
> FENOK(MGAU,MSH,2) , FENIK(MGAU,MSH,2) , FCTOK(MGAU,MSH,2) , 00000800
> FBETA (MSH,MSH,2) , FADJ (MGAU,MSH) , INTR (MGAU,MSH) 00000900
COMMON /TABLE2/           00001000
> FALL (MSH,MSH,2,2) , FQ1 (MSH,2,2) , FQ2 (MSH,2,2) , 00001100
> FALLR (MSH,MSH,2) , FQR (MSH,2) 00001200
DIMENSION ZNUTR3 (MTBL1) , ZTABLE1 (MTBL2) , ZTABLE2 (MTBL3) 00001300
EQUIVALENCE 00001400
> ( ZNUTR3 (1) , CDNI (1) ) , 00001500
> ( ZTABLE1 (1) , FDNO (1) ) , 00001600
> ( ZTABLE2 (1) , FALL (1) ) 00001700
COMMON /TABLP/ MTBL1 , MTBL2 , MTBL3 00001800
```

図41-2 #COMDECK

#### (5) PROGRAM 記述

ユーザが、システムとは異なるモジュールを追加、または置換する場合、ユーザエントリGEM(LIB1D.FORT)にそれらモジュールを、システム記述の規約(「3.4ユーザ専用のソース・モジュールの作成」参照)に従い、作成し登録する。

ここでは、ポロイダル電流密度を計算するモジュール「VJP」を例にとり説明する(図42)。

```

00100 *DECK VJP.USER1
00200     SUBROUTINE     VJP( KER )
00300 C-----
00400 C             POLOIDAL CURRENT DENSITY           I
00500 C-----
00600 C-----
00700 C
00800     INCLUDE DOUBLE
00900     INCLUDE XCZ
01000     INCLUDE YMAG
01100     INCLUDE YBULK
01200     INCLUDE ZMESH
01300 C
01400     DJT(1) = 0.0
01500     DO 300 I = 2, NP1
01600         DJT(I) = (CZP(I)*BPP(I) - DVPS(I))/BZ
01700     300 CONTINUE
01800 C
01900     KER = 0
02000 C
02100     CALL UCMDMP('YMAG ',3,'<<< VJP >>>/.')
02200 C
02300     RETURN
02400     END
02500 /* INCLUDE */
02600     - INC UCMDMP,IN=SYSTEM

```

図 4 2 ユーザエントリモジュール「VJP」

- (a) システムエントリ GEM LIBJT60.FORT のモジュール「VJP」を、ユーザエントリ GEM(LIB1D.FORT) にコピーする。
- (b) DECK 行の GEM 名を、SYSTEM から USER1 に変える (図 4 2 の実線部)。USER1 という GEM 名は、「LB」および「LIBJCL」のパラメータの 1 つである DD 名 (USER (USER1)) と一致させる。
- (c) モジュールの内容を修正する (この例では、内容の修正はない)。
- (d) サブルーチン VJP.USER1 内で CALL されているモジュールが、システムエントリモジュールか、またはユーザエントリモジュールかを確認する。もし、ユーザエントリモジュールであれば、「END 行」の後の「INCLUDE 行」に、その旨を記述する。

```
- INC ユーザエントリモジュール名 , IN=USER1
```

なお、この例ではコモン変数をダンプするルーチン「UCMDMP」が CALL されているだけなので、「INCLUDE 行」に変更はない。

#### (6) データ構造の組上げ

図 4 3 に、TSS マクロコマンド「LIBJCL」を用いて、JCL CDGEN を作成する際の

入力と、端末上の出力例を示す。

- (a) 'LIBJCL CDGEN' と入力する(イタリック体)。  
 (b) デフォルト値の資源情報、およびOPTP文の内容を出力してくる。

```
(a)  LIBJCL CDGEN
    <<< T(1) W(2) C(3) P(0) I(4) OPN >>>
    <<< OPTP PASSWORD=■■■■■■■■■■,MSGLEVEL=(1,1),NOTIFY=J3051 >>>

    ENTER ;; T W C P I OPN NLP      OR 'ENTER-KEY'
(b)  =====> 2 2 3 0 4 CLS
    ENTER ;; MSGLVL NOTIFY          OR 'ENTER-KEY'
(c)  =====>
(d)  KEY = CDGEN
(e)  IF YOU HOPE RE-ENTER KEYWORD =====>
```

図43 LIBJCL CDGEN

続いて、資源情報の内容の変更を尋ねてくる(====>部)。もし変更がなければ、ENTERキーを打鍵する。変更がある場合は、変更しないパラメータ値も含めて全部のパラメータ値を最初から入力する。

(c) 続いて、OPTP文の内容の変更を尋ねてくる(====>部)。入力方法は、資源情報の場合と同じである。この例では、変更がないため、ENTERキーが入力されている。

(d) マクロコマンド'LIBJCL'で作成しようとしているJCLが、「CDGEN」である旨を出力してくる。

(e) CDGEN以外の内容のJCLに変更するか否かを尋ねてくる(====>部)。

JCL CDGENは、単独で用いられなければならないので、ENTERキーを打鍵する。

(f) 端末上に、図44で示される内容のJCL CDGENが出力される。ユーザは、この段階で新ためて内容を検討し、「SUBMIT」する。ここで作成されたJCLは、ワークファイルに「WKJCL.CNTL」名で保存される。

JCL CDGENによる実行結果の出力リストについては、図30を参照のこと。

最後に出力されるコンディションコードが「0」であることを確認する。もし、コンディションコードが「≠0」であれば、誤りを修正して必ず今一度CDGEN処理をすること。その際、前のCDGEN処理によって作成された、最終COMDECKファイル、およびDECKファイルを消却しておかなければならない(図30参照)。各ファイル名は、

- COMDECK.FORT
- DECK.FORT

となっている。最終COMDECKファイル名、およびDECKファイル名は、TSSマクロコマンド「LIBJCL」(図39-2参照)のデフォルト値となっているので、ユーザは必要に応じてこれらのファイル名を変更することができる。

## (7) プログラム構造の組上げ

プログラム構造を組上げるための、JCLキー・パラメータは「OEFL」である(表14参照)。図45にキー・パラメータの入力例と出力例を示す。ロードモジュールを新規に作成する場合は、キー・パラメータ、「OEFL」に続いて、「NEW」を指定しなければならない。これは、原研計算センタにおけるロードモジュール結合編集の制御文(LKEDIT, LKEDUP)

```

00100 //JCLG JOB
00200 // EXEC JCLG
00300 //SYSIN DD DATA,DLM='++'
00400 // JUSER 71973051,TO.HIRAYAMA,0012.100
00500 T.2 W.2 C.3 P.0 I.4 CLS
00600 OPTP PASSWORD=██████████,MSGLEVEL=(1,1),NOTIFY=J3051
00700 //*****
00800 //***** CONSULTING TEL. 5509 *****
00900 //*****
01000 ++
01100 // DD DSN=J2855.LIBJCL.CNTL(CDGEN),DISP=SHR
01200 // DD DATA,DLM=' '
01300 // EXEC CDGEN,LM='J2855.LIBJT60.LOAD',
01400 //     USO='J3051.LIB1D.FORT',
01500 //     COMDECK='J3051.COMDECK.FORT',DECK='J3051.DECK.FORT',
01600 //     SYSTEM='J2855.LIBJT60.FORT',SYSOUT='*',UNIT=TDS,
01700 //     PARAM='DGBL=2,INCL=20,GEMN=200,EXPS=1'
01800 //GEMA.SYSIN DD *
01900 - PUT ◊INC,OUT=SYSUT2,NOID
02000 - PUT ≠SIZE,OUT=SYSUT2,NOINCEXP,AD
02100 - END
02200 /*
02300 //GEMB.SYSIN DD *
02400 - PUT ≠ SYSGEN,OUT=SYSGEN,NOID
02500 - PUT ≠ SIZE,OUT=SYSUT2,NOINCEXP,NOID,AD
02600 - PUT ≠ COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
02700 - END
02800 /*
02900 //GEMC.SYSIN DD *
03000 - PUT ≠ COMDECK,OUT=SYSUT2,NOINCEXP,NOID,AD
03100 - END
03200 /*
03300 //
03400 //

```

図44 JCL CDGEN

の内容が、更新する場合と、新規作成の場合で異なるためである：

新規作成

```

ENTRY MAIN
NAME TEMPNAME

```

更新

```

INCLUDE OLDLM(TEMPNAME)

```

## ENTRY MAIN

NAME TEMPNAME(R)

更新する場合は、「NEW」パラメータを指定する必要はない。

- (a) 「LIBJCL OEFL NEW」と入力する(実線部)。  
 (b) デフォルト値の資源情報、およびOPTP文の内容を出力してくる。  
 (c) 続いて、資源情報の内容の変更を尋ねてくる(====>部)。

```
(a) LIBJCL OEFL NEW
(b) <<< T(1) W(2) C(3) P(0) I(4) OPN >>>
    <<< OPTP PASSWORD=LIBRARY,MSGLEVEL=(1,1),NOTIFY=J3051 >>>
    $$ ENTER ;; T W C P I OPN NLP OR 'ENTER-KEY'
(c) =====> 3 2 4 0 4 CLS
    $$ ENTER ;; MSGVLV NOTIFY OR 'ENTER-KEY'
(d) =====>
(e) KEY = OEFL
(f) $$ IF YOU HOPE MODIFICATION THEN ENTER KEYWORD =====>
    $$ ENTRY SOURCE FILE ???
    IF SYSTEM FILE, THEN INPUT 'S'
    IF USER FILE, THEN INPUT 'U'
(g) ==> U
    FILE NAME 'J3051.LIB1D.FORT' ?? THEN ENTER 'RETURN-KEY'
    ELSE ENTER FILE NAME (FULL NAME)
(h) =====>
    $$ ENTER MODULE NAME OR 'MODNAME'
    AND INCLUDE('I'), NOINCLUDE('N'), 'ENTER-KEY'
(i) =====> VJP N
    **** PUTTING FROM SYSTEM FILE ****
    $$ ENTER MODULE NAME OR 'MODNAME'
    IF PUT NONE THEN 'ENTER-KEY'
(j) =====> YMAIN I
```

## 図45 LIBJCL OEFL

T = 3, C = 4, および出力先を 'CLS' に変更する。

- (d) OPTP文の内容の変更を尋ねてくる(====>部)。

変更がなければ、ENTER キーを打鍵する。

- (e) マクロコマンドLIBJCLで作成しようとしているJCLが、OEFLである旨を出力してくる。  
 (f) OEFL以外の内容のJCLに変更するか否かを尋ねてくる(====>部)。例えば、STANユーティリティを使用するためのJCLに変更する場合は、OESと入力すればよい。この例では、JCLの内容に変更がないので、ENTER キーを打鍵する。  
 (g) コンパイルおよびリンク(OEFL)の対象となるソースファイルが、システムのファイルか、あるいはユーザのファイルのいずれかである旨を指定する。システムファイルをソースファイルとする場合は、「S」、ユーザファイルであれば、「U」と入力する。「U」と入力すると、ユーザファイルの指定の後、再度システムファイルを指定することができる。最

初に、「S」と入力すると、ユーザファイルをコンパイル、リンクの対象とすることはできない。ここでは、「U」と入力する。

(h) コンパイル、リンクの対象となっているユーザ・ソースファイル名を出力する。ファイル名に変更がなければ、ENTER キーを打鍵する。変更があれば、そのファイル名を完全修飾付名で入力する。

(i) ソースファイルの中から、ユーザ・エントリ・モジュール名を入力する。この時、エントリ・モジュールについて、展開モード、或は非展開モードの指定を必ず付加しなければならない。ここでは、ユーザ・エントリ・モジュールとして「VJP」を、非展開モードで入力している。なお、複数のエントリ・モジュールが在る場合は、全体を引用付で括り入力する：

```
'ABC, DEF, GHI', N
```

(j) 最初に、「U」を指定したので、再度システムファイルを指定するか否か尋ねてくる。指定しない場合は、ENTER キーを打鍵すればよい。指定する場合は、ユーザ・ソースファイルの方法と同じである。ここで、注意すべきことは、最初にロードモジュールを作成する時は、必ずエントリ・モジュールの「YMAIN」ルーチンを、展開モードで指定しなければならない。

端末上に、図46で示される内容のJCL OEF Lが出力される。ここで作成されたJCLは、WKJCL.CNTL名で、ワークファイルに保存される。

```
0100 //JCLG JOB
0200 // EXEC JCLG
0300 //SYSIN DO DATA,DLM='++'
0400 // JUSER 71973051,TO,HIRAYAMA,0012.100
0500 T.3 W.2 C.4 P.O I.4 CLS
0600 OPTP PASSWORD=██████████,MSGLEVEL=(1,1),NOTIFY=J3051
0700 //*****
0800 //***** CONSULTING TEL. 5509 *****
0900 //*****
1000 ++
1100 // DD DSN=J2855.LIBJCL.CNTL(ORG),DISP=SHR
1200 // DD DSN=J2855.LIBJCL.CNTL(EOS),DISP=SHR
1300 // DD DATA,DLM='$$'
1400 //*****
1500 //*** ORG ***
1600 //*****
1700 // EXEC ORG,GEMSO='J3051.LIB1D.FORT',SYSTEM='J2855.LIBJT60.FORT',
1800 // SYSGEN='J3051.LIB1D.FORT',
1900 // DDN1=USER1,DDN2=USFR2,DDN3=USER3,
2000 // US01='J3051.LIB1D.FORT',US02='J3051.LIB1D.FORT',
2100 // US03='J3051.LIB1D.FORT',
2200 // PARAM='DBGL=1,INCL=20,GEMN=200,EXPS=1'.
2300 // SYSOUT='*',LM='J2855.LIBJT60.LOAD'
2400 //GEMA.SYSIN DD *
2500 - PUT #SYSGEN,OUT=SYSGEN,NOID
2600 - END
2700 /*
2800 //GEMB.SYSIN DD *
2900 - PUT @NOINC,OUT=SYSUT2,NOID
```

次頁へ続く

```

3000 - END
3100 /*
3200 //GEMC.SYSIN DD *
3300 - GROUP GG,MODNAME=(VJP)
3400 - PUT GG,OUT=SYSUT2,NOINCEXP,NOID,AD
3500 - END
3600 /*
3700 //GEMD.SYSIN DD*
3800 - PUT @INC,OUT=SYSUT2,NOINCEXP,NOID,AD
3900 - GROUP GG,MODNAME=(YMAIN)
4000 - PUT GG,OUT=SYSUT2,NOINCEXP,NOID,AD
4100 - END
4200 /*
4300 //*****
4400 /**  EOS      ***
4500 //*****
4600 // EXEC EOS,LM='J2855.EOS23.LOAD',COMDECK='J3051.COMDECK.FORT',
4700 //      DECK='J3051.DECK.FORT',
4800 //      PARAM='SLIST=0,LINECM=1',SYSOUT='*'
4900 //*****
5000 /**  FORTGE   ***
5100 //*****
5200 //FORT EXEC FORTGE,SO='&&EOSOUT',A='NOSOURCE,NOISN,NOMAP',B='PRINT',
5300 //      ELVL=W,NAME=NONAME,CALL=BYNAME,TRBK=GOSTMT,OBJS='30,10',
5400 //      Q='',SYSOUT='*',STATIS=NOSTATIS
5500 //*****
5600 /**  LKEDUP   ***
5700 //*****
5800 //LKEDUP EXEC LKEDUP,LM='J3051.LIB1D',Q='.LOAD',A='NOMAP',
5900 //      B='LIST',WRKS='100,50',UPS=30,CNTL=NO,
6000 //      PRVLIB='SYS9.NO',GGG='SYS9.GGS',USER=OFF,CSPC='100,50,10',
6100 //      CUNIT=TSSWK,SYSOUT='*'
6200     ENTRY MAIN
6300     NAME TEMPNAME
6400 /*
6500 6500
6600 //

```

図 4 6 JCL OEFL

## (8) 端末上でのユーザ・エントリ GEM の修正、およびロードモジュールの更改

端末上で、ユーザ・エントリ GEM、およびロードモジュールを修正するための TSS マクロコマンドが、「LB」である。このコマンドを用いて、ロードモジュールを新規に作成することもできるが、コンパイルおよびリンクに時間がかかるため、原則として、「LB」コマンドはロードモジュールの更改だけを行うものとする。「LB」コマンドの使用に当っては、パラメータをユーザ専用の値に設定しておくことが必要である（6.3 節を参照）。

図 4 7 に示す例は、ユーザ・エントリ GEM のモジュール「NGPLS」をリンクした例である。

(a) TSS コマンド 'LB' を入力する。一日の最初にする場合は、続けて 'FILE' と入力する。このパラメータによって、編集に必要な作業用ファイルが割当てられる。

(b) ユーザ・エントリ GEM (ここでは, J3051.LIB1D.FORT) のセッションが開かれる。モジュールを修正する場合は, 「UPDATE」モードにする。

修正が終了すれば, 'END' を入力する。

(c) 「ORG」処理をするユーザ・エントリ・モジュールを入力する。その際必ず, 展開モード('I')または, 非展開モード('N')を指定する。ここでは, モジュール「NGPLS」

(a) LB FILE

```
<<< LB START 10/27/82 11:41:44 >>>
UNDER ALLOCATION WORK FILE
MODULE '#SYSGEN' HAS BEEN PUT.
$$ GEM3 'J3051.LIB1D.FORT' START. WHEN UPDATE END, ENTER 'END'
THE GROUP 'A' WAS DEFINED.
GEM3
```

(b) END

```
$$ ENTER MODULE NAME OR 'MODNAME'
INC MODE ; 'I' NOINC MODE ; 'N' OR NONE
IF YOU HOPE EXIT THEN ENTER 'ENTER-KEY'
```

(c)

```
====> NGPLS N
MODULE '@NOINC' HAS BEEN PUT.
THE GROUP '@AA' WAS DEFINED.
MODULE 'NGPLS' HAS BEEN PUT.
$$ ENTER MODULE NAME OR 'MODNAME'
INC MODE ; 'I' NOINC MODE ; 'N' OR NONE
IF YOU HOPE EXIT THEN ENTER 'ENTER-KEY'
```

(d)

```
====>
==== END OF MODULES PUTTING ( FILE : J3051.LIB1D.FORT ) ====
***
$$ IF PUT MODULE OF OUTER FILE THEN ENTER DSN (FULL NAME)
```

(e)

```
IF YOU HOPE END OF PUTTING THEN ENTER 'ENTER-KEY'
====>
```

(f)

```
<<< ORG START >>>
CALL 'J2855.LIBJT60.LOAD(ORG)' 'DBGL=2,INCL=20,GEMN=100,EXPS=1'
** OPTLBL ** (NGPLS DECK USER1 )
1 NBI 0
2 HY 1
3 AA 0
4 BB 0
5 HY1 0
6 HE 0
7 NBI1 0
8 NBI2 0
** PTTBLT ** (NGPLS DECK USER1 )
<< SYSTEM GENERATE >>
*SYSGEN #SYSGEN LIB1D
=== HPR, JZ, TPR-TE === (82/6)
***
*MODULE LABEL
MODULE GEM
1 NGPLS DECK USER1
<<< ORG END ( CODE=0 ) >>>
FILE FT09FOO1 NOT FREED, IS NOT ALLOCATED
```

次頁へ続く

```

*****
::::: COMDECK FILE = 'J3051.COMDECK.FORT'
::::: DECK FILE   = 'J3051.DECK.FORT'
*****
(g) $$ IF YOU HOPE CHANGING FILE-NAME THEN ENTER 'C' ==>

(h)          <<<  EOS  START  >>>
CALL 'J2855.EOS23.LOAD(EOS23)' 'LINECM=1,SLIST=0'
          <<<  EOS  END  ( CODE=0 )  >>>
$$ IF YOU HOPE <<< COMPILE >>> THEN ENTER 'C'
      .....          EXIT          THEN ENTER 'O'
      .....          RE-TRY ORG    THEN ENTER 'NOT C'
(i) ==> C
$$ COMPILER START
(j) $$ IF YOU HOPE COMPILE-LIST THEN ENTER 'S' ==>
FORT PSW1.FORT FIXED NOGO NOSEQ NOSOURCE DBJ(PSW1.OBJ) PRINT(*)
***
          <<<  COMPILE O.K.  ( CODE=0 )  >>>>
$$ IF YOU HOPE <<< LINKAGE >>> THEN ENTER 'L'
      .....          EXIT          THEN ENTER 'O'
      .....          RE-TRY GEM3 MODE THEN ENTER 'NOT L'
(k) ==> L
¥¥ IF YOU HOPE CONDENSE ??? THEN ENTER 'C'
          ----- ELSE ENTER 'ENTER-KEY'

(l) ==>
(m) ALLOC DA('J3051.LIB1D.LOAD') F(FLOAD) SHR
LINK (PSW1.OBJ 'J2855.#SYSDATA.DATA(STAB)') F LOAD('J3051.LIB1D.LOAD') PRINT(*)
FACOM OSIV/F4 LINKAGE EDITOR VO3L16 DATE 82.10.27 TIME 17.29.10
OPTIONS SPECIFIED - NONE
GENERATED - NOXREF,NOMAP,NOLIST
***VALUES IN EFFECT*** SIZE< DEFAULT USED >=(251904,83968)
MAX. LENGTH OF OUTPUT TEXT BLOCK = 12288
**MEMBER NAME** TEMPNAME NOW REPLACED IN LIBRARY.
**TTR**( 92 / 06 - 187 / 05 ) **AUTHORIZATION CODE**( 0 )
**NOW 0 TRACK(S) LEFT UNUSED IN LIBRARY CONVERING 6 EXTENT(S).
FREE DA('J3051.LIB1D.LOAD')
$$ LINKAGE END. NOW J3-51.LIB1D.LOAD UPDATED ( CODE=0 )
***** CONGRATURATION *****

READY

```

図 4 7 T S S コマンド L B

を、非展開モードで入力している。

- (d) ユーザ・エントリ・モジュールが複数個ある場合は、最下層のモジュールから、順次、非展開モードで入力する。入力が終了すれば、「ENTER」キーを打鍵する。
- (e) 最初に指定したユーザ・エントリ GEM のセッションが終了する。他のユーザ・ソースファイルを、エントリ GEM とする場合は、完全修飾詞付ファイル名を入力する。入力が必要ならば、「ENTER」キーを打鍵する。
- (f) 「ORG」処理
- (g) COMDECK ファイル、および DECK ファイルを、最初に指定したファイルから変更する場合は、「C」と入力し、変更ファイル名を完全修飾詞付で入力する。変更が必要ならば、

「ENTER」キーを打鍵する。

(h) 「EOS」処理

コンディションコード，「CODE = 0」を確認。

(i) コンパイルをする場合は，「C」を入力する。再度「ORG」処理をする場合は，「C」以外のアルファベットを入力する。「LB」による一連の処理を打切る場合は，「0」を入力すると，「READY」モードに戻る。ここでは「C」を入力する。

(j) コンパイル・リストを，端末上に出力するか否かを尋ねてくる。出力させる場合は，「S」を，そうでない場合は「ENTER」キーを入力する。

コンパイルのコンディションコード，「CODE = 0」を確認。

(k) リンクするか否かを入力する。

リンクする場合は，「L」を，また「LB」による処理を打切る場合は，「0」を入力する。エラーの発生等の理由で，「GEM3」モードに戻す場合は，「L」以外のアルファベットを入力する。

(l) ロードモジュールをコンデンスするか否かを入力する。コンデンスする場合，「C」を，しない場合は「ENTER」キーを入力する。

(m) コンデンス情報およびリンク情報の表示。

## 9. 結 言

JT-60の完成を控え、実験データ解析コード整備の一環として、FORTRANソースを管理するライブラリ・システムの開発を行った。このシステムは、「JT-60実験解析コード作業グループ報告書」<sup>6)</sup>に基づき、発展性、互換性のあるコード開発を支援することを主眼としたものである。本システムを利用することにより、既開発の1次元トカマク輸送コードは、ソース・ライブラリ・システムとして使用することが可能になった。これにより、ユーザはトカマク輸送コードをブラックボックスとして使用でき、また種々のバリエーションを容易に製作できる。

現時点では、トカマク・シミュレーション・コードの基本構造部分が、システムとして完成している。今後は、汎用シミュレーション・コード、種々の実験データ解析コード群を、本システムを用いて開発していく予定である。

### 謝 辞

ライブラリ・システムの開発に際しては、渡辺典孝、松下 樹、塩尻常晴各氏（数理技研株式会社）の努力に負う所が大きく、ここに記して感謝に代えます。EOSシステムの使用を許可して頂いた、核融合研究部 理論解析研究室 竹田辰興室長に深く感謝いたします。1次元トカマク輸送コードを、ライブラリ・システム化するに当っては、杉原正芳氏の有益な助言が、大きな力となった事を記し感謝します。実験データ解析コード整備計画全般を通じての、大型トカマク開発部 苫米地 頭部長、吉川允二次長、計画室 平岡 徹室長の御激励に感謝します。

### 参考文献

- (1) M.SUGIHARA, private comunication  
T.HIRAYAMA, private comunication
- (2) FACOM OSN/F4 GEM3 使用手引書
- (3) 竹田辰興, 常松俊秀, 栗田源一 可変配列サイズ・プログラムのためのプリプロセッサ・システム「EOS」, JAERI-M 82-097 (1982)
- (4) ORG プログラム・マニュアル
- (5) CDGEN プログラム・マニュアル
- (6) 解析コード作業グループ JT-60実験解析コード作業グループ報告書 所内資料

## 9. 結 言

JT-60の完成を控え、実験データ解析コード整備の一環として、FORTRANソースを管理するライブラリ・システムの開発を行った。このシステムは、「JT-60実験解析コード作業グループ報告書」<sup>6)</sup>に基づき、発展性、互換性のあるコード開発を支援することを主眼としたものである。本システムを利用することにより、既開発の1次元トカマク輸送コードは、ソース・ライブラリ・システムとして使用することが可能になった。これにより、ユーザはトカマク輸送コードをブラックボックスとして使用でき、また種々のバリエーションを容易に製作できる。

現時点では、トカマク・シミュレーション・コードの基本構造部分が、システムとして完成している。今後は、汎用シミュレーション・コード、種々の実験データ解析コード群を、本システムを用いて開発していく予定である。

### 謝 辞

ライブラリ・システムの開発に際しては、渡辺典孝、松下 樹、塩尻常晴各氏（数理技研株式会社）の努力に負う所が大きく、ここに記して感謝に代えます。EOSシステムの使用を許可して頂いた、核融合研究部 理論解析研究室 竹田辰興室長に深く感謝いたします。1次元トカマク輸送コードを、ライブラリ・システム化するに当っては、杉原正芳氏の有益な助言が、大きな力となった事を記し感謝します。実験データ解析コード整備計画全般を通じての、大型トカマク開発部 苫米地 頭部長、吉川允二次長、計画室 平岡 徹室長の御激励に感謝します。

### 参考文献

- (1) M.SUGIHARA, private communication  
T.HIRAYAMA, private communication
- (2) FACOM OSN/F4 GEM3 使用手引書
- (3) 竹田辰興, 常松俊秀, 栗田源一 可変配列サイズ・プログラムのためのプリプロセッサ・システム「EOS」, JAERI-M 82-097 (1982)
- (4) ORG プログラム・マニュアル
- (5) CDGEN プログラム・マニュアル
- (6) 解析コード作業グループ JT-60実験解析コード作業グループ報告書 所内資料

## 9. 結 言

JT-60の完成を控え、実験データ解析コード整備の一環として、FORTRANソースを管理するライブラリ・システムの開発を行った。このシステムは、「JT-60実験解析コード作業グループ報告書」<sup>6)</sup>に基づき、発展性、互換性のあるコード開発を支援することを主眼としたものである。本システムを利用することにより、既開発の1次元トカマク輸送コードは、ソース・ライブラリ・システムとして使用することが可能になった。これにより、ユーザはトカマク輸送コードをブラックボックスとして使用でき、また種々のバリエーションを容易に製作できる。

現時点では、トカマク・シミュレーション・コードの基本構造部分が、システムとして完成している。今後は、汎用シミュレーション・コード、種々の実験データ解析コード群を、本システムを用いて開発していく予定である。

### 謝 辞

ライブラリ・システムの開発に際しては、渡辺典孝、松下 樹、塩尻常晴各氏（数理技研株式会社）の努力に負う所が大きく、ここに記して感謝に代えます。EOSシステムの使用を許可して頂いた、核融合研究部 理論解析研究室 竹田辰興室長に深く感謝いたします。1次元トカマク輸送コードを、ライブラリ・システム化するに当っては、杉原正芳氏の有益な助言が、大きな力となった事を記し感謝します。実験データ解析コード整備計画全般を通じての、大型トカマク開発部 苫米地 頭部長、吉川允二次長、計画室 平岡 徹室長の御激励に感謝します。

### 参考文献

- (1) M.SUGIHARA, private comunication  
T.HIRAYAMA, private comunication
- (2) FACOM OSN/F4 GEM3 使用手引書
- (3) 竹田辰興, 常松俊秀, 栗田源一 可変配列サイズ・プログラムのためのプリプロセッサ・システム「EOS」, JAERI-M 82-097 (1982)
- (4) ORG プログラム・マニュアル
- (5) CDGEN プログラム・マニュアル
- (6) 解析コード作業グループ JT-60実験解析コード作業グループ報告書 所内資料