

[04_03]九州大学大型計算機センター広報 : 4(3)

<https://doi.org/10.15017/1467976>

出版情報 : 九州大学大型計算機センター広報. 4 (3), pp.1-54, 1971-06-26. 九州大学大型計算機センター
バージョン :
権利関係 :

ジョブ制御マクロ一覧

1. ジョブの始めを示すための制御文

NO

1	2 3	4	オペランド
\$	NO		<受付番号>

機能：（１）ジョブの開始の指示

（２）これ以後のカードがEL型で穿孔されていることを示す。

パラメータの意味

<受付番号> センターと利用者間の連絡の際、ジョブの識別番号になります。

〔6桁の数字〕

上のカードは計算依頼カードになっており受付で利用者が入れます。

2. ジョブの性質を示すための制御文

QJOB

1	2 3 4 5	6	オペランド
\$	QJOB		<課題番号>、<登録名>、<ジョブ種別>、<使用言語> [, TIME=mm.ss][, LP=n][, LPL=n][, CP=n] [, CORE=nk][, CONDITION=(下限、上限)]

機能：ジョブの性格を決めます。

パラメータ：

<課題番号> }
<登録名> } 課題登録申請書に記入されるように書かねばなりません。

<ジョブ種別> X、A、B、C、Dのいずれかを記入します。

<使用言語> 主要な使用語を記入してください。

パラメータ	記入したとき(注1)	省略したとき
TIME=mm.ss	このジョブの打ち切り時間をmm分ss秒とする。 $0 < mm \leq 32$ でこれを越えるとDジョブになります。	各ジョブ別に与えられる最大値は Xのとき: 30秒 Aのとき: 1分 Bのとき: 8分 Cのとき: 32分
LP=n	このジョブの印刷枚数をnページで打切る。 $0 < n \leq 600$ でこれを越えるとDジョブになります。	各ジョブ別に与えられる最大値は X : 30ページ A : 50ページ B : 120ページ C : 600ページ
LPL=n	このジョブの印刷行数をn行で打切る。 $0 < n < 30,000$ でこれを越えるとDジョブになります。(注2)	各ジョブ別に与えられる最大値は X : 1,500 A : 2,500 B : 6,000 C : 30,000
CP=n	このジョブ出力カード枚数をn枚で打切る。 $0 \leq n \leq 5,000$ でこれを越えるとDジョブになります。	各ジョブ別に与えられる最大値は X : 0枚 A : 0 B : 1,000 C : 5,000
CORE=nK	このジョブのコア使用量の最大値(実行時)を指定する。この指定により、そのジョブ種別の制限(別表参照)までCPU時間を延長できる。(注3)	64KWとなる
CONDITION =(下限, 上限)	$0 \leq \text{下限} < \text{上限} \leq 511$ 下限、上限を上条件をみたす3桁の数字で指定する。 下限 \leq 完了コード \leq 上限となるジョブステップがあればジョブはそのステップまでで終る。	CONDITION=(470,511)

(注1) この指定が各ジョブ種別の制限を越えた場合には、その種別で許される最大値で自動的におきかえる。

例) \$ QJOB ..., A, LP=100 LP=50におきかえられる

(注2) LP, LPLのどちらか先に打ち切りに達した方で出力が打切られる。

(注3) コア使用量と、CPU時間は本広報52ページ表1を見てください。

3. 翻訳のためのジョブ制御マクロ

① FORTRAN

1	2 3 4 5 6 7 8	9
	命 令	オ ペ ラ ン ド
\$	FORTRAN	[GO][,NOLIST][,MAP][,NOOPT][,LMAP][,SEQ] [,MACROSTEP=n][,COMPUNIT=n] [,ELMNAME=(メンバ名,...)][,OUTPUT=REMOTE]

機能：この制御マクロの後に続くFORTRANのプログラムをコンパイルする。

パラメータの説明：

パラメータ	記 入 し た と き	省 略 し た と き
GO C1)	コンパイルの完了コードが470未満のものについて、コンパイラ自身がLIEDを呼び出して処理をさせる。この後の制御文は必ず\$GOでないといけない。	コンパイルのみする。
NOLIST	ソースプログラムのリストを出さない。 コンパイルエラーメッセージのみ出す。GOを指定した時にはLIEDのリストもとらない。エラーメッセージのみ出す。	ソースプログラムのリストとエラーメッセージを出す。 GOを指定した時にはLIEDのリストのエラーメッセージも出す。
MAP	コンパイル時の定数、変数等のマップ(番地割り付け)を出す。	マップを出さない。

パラメータ	記 入 し た と き	省 略 し た と き
NOOPT	C 2) optimizeしないコンパイルをする。	optimizeしたコンパイルをする。
OPT 1	配列のみoptimizeしたコンパイルをする。	
L MAP	GOを指定した時にL I E Dのコ アマップ(実行時の番地割り付け) を出す。	GOを指定した時にL I E Dのコ アマップを出さない。
SEQ	ソースプログラムのカードシーク エンス(第73~80桁)をチェックす る。エラーがあれば*を打ち出す。	カードシークエンスのチェックを しない。
MACROSTEP = n	\$FORTRAN を複数個続けて挿 入する時に異なった番号(数字3 けたまで)を与える。	番号はなにもつかない。
COMPUNIT = n	in core で同時にコンパイルする プログラム単位の個数をnとする。	n = 5
C 4) ELMNAME (メンバ名,,...)	磁気テープや大記憶にL I B Eで 作成したソースプログラムファイ ルの中から指定したメンバのみコ ンパイルする。	ファイルの全メンバをコンパイル する。
OUTPUT =REMOTE	コンパイル時のシステム出力装置 が端末となる。	センタのラインプリンタとなる。

C1)

(1) FORTRANでコンパイルしたものを結合、実行させるのに3つの方法がある。

1. LIEDで結合できるRBはFORTRAN基本外部関数のみ(注1)

(詳しくはGOマクロ参照)

\$NO	
\$QJOB	
\$FORTRAN GO	\$FORTRAN
<div style="border: 1px solid black; padding: 2px; display: inline-block;">ソース</div>	:
\$GO	\$GO
<div style="border: 1px solid black; padding: 2px; display: inline-block;">データ</div>	
\$JEND	の指定はできない。

2. LIEDで結合できるRBは、基本外部関数、SSL、センターが登録したRB

```

$NO
$QJOB
$FORTRAN


ソース


$LIEDRUN


データ


$JEND

```

3. LIEDで結合できるRBは2の他に利用者の専用ファイルのRB

```

$NO
$QJOB
$FORTRAN


ソース


$LIED


LIED制御文


$ファイル定義マクロ
$RUN


データ


$JEND

```

LIDEの項を参照してください。

注1) 1、2、3の内1でやれるジョブは1でやった方が負担金も安くなり、返却も早くなる。

C 2) optimize の功罪

利点

1. 実行速度が速いので、コンパイル時間に比して実行時間の長いジョブRB、EBで実行するジョブはoptimize した方がいい。

欠点

1. コンパイルの速度が遅い。
2. コンパイルの時、ワークエリアをよけいにとる (注2)
3. デバッグ文が使えない。
4. エラー個所とI. S. N. の対応がとれない。

(注2) ・CPU時間の長短から考えた場合コンパイラー結合ー実行のジョブでは、このtotalのCPU時間が短い方を選ぶべきである。

- ・カード枚数が多いプログラムはCOMPUNIT=1でやった方がよい。
- ・コンパイル時にワークエリアがとれないというメッセージが出たときは、すみやかに相談員に申し出てください。

C 3) MACROSTEP=n

```
$NO
```

```
$QJOB
```

```
$FORTRAN
```

```
$FORTRAN MACROSTEP=1
```

```
$FORTRAN MACROSTEP=n-1
```

```
$FORTRAN GO, MACROSTEP=n or $FORTRAN
```

```
MACROSTEP=n
```

```
$GO
```

```
$LIEDRUN
```

C 4) 共用ボリューム上のソースファイルからコンパイルする場合

```
$FORTRAN GO, ELMNAME=(A, B, C) A, B, Cというメンバのみ
```

```
$FD FORTRAN [MACROSTEPのn]-SOURCE, FILE=(...)
```

```
$GO
```

C 5) NOLIST, MAP, SEQ

これらの指定はOPTION文でもできる。(詳しくはFORTRAN解説編II参照)

② ALGOL

1	2 3 4 5 6	7	
	命 令		オ ペ ラ ン ド
\$	A L G O L		[NOLIST],[HTYPE],[SEQ],[MACROSTEP=n] [,ELMNAME=(メンバ名,...)][,OUTPUT=REMOTE]

機能：この制御マクロの後に続くALGOLのプログラムをコンパイルする。

パラメータの説明

パラメータ	記 入 し た と き	省 略 し た と き
C 1) N O L I S T	ソースプログラムのリストを出さない。エラーメッセージのみ出す。	ソースプログラムのリストもエラーメッセージも出さない。
H T Y P E	ALGOLソースプログラムが026コード系で記述されている。	C 2) 029コード系で記述されている。
C 1) S E Q	ソースプログラムのカードシーケンス(第73桁~80桁)をチェックする。	カードシーケンスをチェックしない。
M A C R O S T E P = n	\$ALGOLを1ジョブで複数個使用する時異なった番号(3桁まで)を与える。	番号はなにもつかない。
C 3) E L M N A M E =(メンバ名,...)	磁気テープや大記憶にLIBEで作成したソースプログラムファイルの中から指定したメンバのみコンパイルする。	ファイルの全メンバをコンパイルする。
O U T P U T =REMOTE	コンパイル時のシステム出力装置が端末となる。	センタのラインプリンタとなる。

- C 1) このパラメータの変更は制御データ (F 2 3 0 - 6 0 ALGOLプログラミング解説編「制御データ」の項参照)
- C 2) F 2 3 0 - 6 0 ALGOLプログラミング解説編「基本事項」の区切り記号の項を参照してください。
- C 3) 共用ボリューム上のソースファイルからコンパイルする場合
FD文の混ぜ合わせにより

```

$ALGOL ELMNAME=(A,B) ELMNAMEの指定をしなければ全メンバ
$FD ALGOLn-SOURCE, FILE=(...)をコンパイルする。
      ↑
      MACROSTEPのn
    
```

③ F A S P

1	2 3 4 5	6	
	命 令		オ ペ ラ ン ド
\$	F A S P		{LIST}{,DEBUG}{,NOSEQ}{,MACROSTEP=n} {,ELMNAME=(メンバ名,...)}{,OUTPUT=REMOTE}

機能：この制御マクロの後に続くFASPソースカードをアSEMBルする。

パラメータの説明：

パラメータ	記 入 し た と き	省 略 し た と き
L I S T	FASPソースプログラムのリストとエラーメッセージを出す。	エラーメッセージのみ出す。
D E B U G	ソースプログラムで第1桁が/ (スラント) のものも含めてアSEMBルし、相対形式プログラムとそ の中で使用された番地記号の記号 テーブルを作成する。	第1桁が/のものは無視してアSEMBルし、相対形式プログラム本体だけを作成する。
N O S E Q	ソースプログラムのカードシーク エンス (第73~80桁) をチェック しない。	カードシークエンスをチェックを する。

パラメータ	記 入 し た と き	省 略 し た と き
MACROSTEP = n	\$FASPを1ジョブで複数個使用する時異なった番号(3桁まで)を与える。	番号はなにもつかない。
C 1) ELMNAME =(メンバ名, ...)	磁気テープや大記憶にLIBEで作成したソースプログラムファイルの中から指定したメンバのみアセンブルする。	ファイルの全メンバをアセンブルする。
OUTPUT =REMOTE	アセンブル時のシステム出力装置が端末となる。	センタのラインプリンタとなる。

C 1) ファイルからアセンブルする場合

\$FASP ELMNAME=(X, Y) 全メンバの時には指定してはいけない。

\$FD FASP (MACROSTEPで指定したn) -SOURCE, FILE= ()

\$LIEDRUN

C 2) FORTRANとの結合

\$GO使用の場合

\$FASP

\$FORTRAN GO

\$GO

\$LIEDRUNの場合

\$FORTRAN \$FASP

\$FASP \$FORTRAN

\$LIEDRUN \$LIEDRUN

4. 結合編集のためのジョブ制御マクロ

LIED

1 2 3 4 5 6

	命 令		オ ペ ラ ン ド
\$	L I E D		[NOLIST][LMAP][MACROSTEP=n] [,OUTPUT=REMOTE]

機能: LIED (結合編集プログラム) の制御文をこのマクロの後に書くことにより、さまざまな相対形式プログラムをまとめて、実行形式プログラムを作成することができる。

(専用ファイルのRBを組み込む利用者は必ず使用しないとけない。)

パラメータの説明:

パラメータ	記 入 し た と き	省 略 し た と き
NOLIST	LIEDのリストとエラーメッセージを出さない。	リストとエラーメッセージを出す。
L MAP	作成された実行形式プログラムのコマアップ(主記憶上での番地割り付け)を出す。	コマアップを出さない。
MACROSTEP = n	SLIEDを1ジョブで複数個使用する時異なった番号(3桁以内)を与える。	番号を与えない。
OUTPUT = REMOTE	LIED時のシステム出力装置が端末となる。	センタのラインプリンタとなる。

C1) LIEDの制御文に関して詳しくはF230-60 LIED文法編を参照してください。共用ボリームの専用ファイル(ファイル定義名PRIVLIB, ファイルQU.RB.00010)を組み込む例

```

$LIED      LMAP
NAME       EXQTPRGM注1),      ENTRY=ELM(PRG.MAIN)注2)
           [DOMAIN OVLY]注3)
CALL       SYSLIB注4)
SGMT       SEG1
SELECT     RELBIN, PRIVLIB(ELMA)注5)
FIN
$DAFILE   FDNAME=PRIVLIB,FILENAME=QU.RB.00010注6)
$RUN
    
```

注1) \$RUNのEBNAMEというパラメータで与えるものは実行形式プログラム名を示す。省略するとEXQTPRGMとしている(\$RUN参照)のでEXQTPRGMとしなければなりません。

- 注2) 実行開始プログラム名を記入します。
FORTRAN, ALGOLの主プログラムから開始する時にはPRG. MAINとしなければなりません。
- 注3) オーバレイ構造の場合書きます。
- 注4) FORTRANライブラリをCALLする時、その他必要に応じてCALLするファイルとファイル定義名は下表の通り(センター共用ファイル)

フ ァ イ ル	ファイル定義名
FORTRANライブラリ	SYSLIB
ALGOLライブラリ	SYSLIB1
FORTRAN SSL	F. SSLIB
ALGOL SSL	A. SSLIB
センターに登録したライブラリ	P. LIB
センターのライブラリ	Q. LIB

Q. LIBにはPTRのライブラリ, QDLIST等が入っています。

例) FORTRANライブラリ、FORTRAN.SSLのあるエレメントを組み込みたい場合。

```

:
CALL      SYSLIB, F. SSLIBとすると自動的に組み込まれます。
:
FIN

```

注5) RELBINファイルはこのジョブステップ以前にFORTRANコンパイラで作られたRB(相対形式)プログラムが入っています。

LIEDはRELBINファイルより必ず組み込みを始めるので、必ず指定しなければなりません。従って専用ファイルに登録したRBのみでLIEDを使用するには

```

$LIED      MACROSTEPのn
           ↓

```

```

$FD LIEDn-RELBIN, FILE=( )として専用ファイルを定

```

義しなければなりません。

LIED 制御文

```

$RUN

```

SELECT文, DELETE文

- (1) SELECT fdname fdnameで示されるファイルの全RBが組み込まれます。
- (2) SELECT fdname (エレメント, ..)指定したエレメントが組み込まれます。
- (3) SELECT fdname fdnameで示されるファイルのRB組み込みを取りやめる。
DELETE fdname
- (4) SELECT fdname fdnameで示されるファイルの中で指定し
DELETE fdname (エレメント, ..)たエレメントを除くRBが組み込まれる。

注6) L I E Dの制御文で示したfdnameは例のように必ずマクロでfdnameとfilenameを指定する必要があります。

5. 実行のためのジョブ制御マクロ

① GO

1	2 3	4
命 令		オ ペ ラ ン ド
S	G O	(EBNAME=実行形式プログラム名)[, STARTPROGRAM=実行開始プログラム名][, LP= n][, TIME= n][, ERR= n][, DATA=データ名][, MACROSTEP= n][, OUTPUTPUT= REMOTE]

機能: \$FORTRAN GOでコンパイル、結合を済ましたジョブの実行のみをする。

パラメータの説明:

パラメータ	記 入 し た と き	省 略 し た と き
EBNAME = 実行形式プログラム名	指定した名前が実行形式プログラムにつけられる。	EXQTPRGMとなる。
STARTPROGRAM = 実行開始プログラム名	指定した名前をもつプログラム単位から実行が開始される。	PRG. MAINとなりFORTRANの主プログラムから実行が開始される。
LP = n	実行時の出力ページ数をn枚で打切る。nはジョブ種別の打ち切り以下でないといけない。	ジョブのトータルの出力が、対応する種別の打ち切り定数で打切られる。

パラメータ	記入したとき	省略したとき
TIME = n	実行時のCPU時間をn秒で打切る。 nはジョブ種別の打切り以下でないといけない。	ジョブのトータルのCPU時間が 対応する種別の打切り定数で打切られる。
ERR = n	実行時のエラーの個数がn1になったらジョブは打切られる。	n = 50となる。
DATA = データ名	磁気テープ、大記憶にLIBEで作成したデータ名を指定する。この時カードからのデータは入力できない。	空となる。
MACROSTEP = n	\$FORTRAN GO \$GO という2つのマクロを同一ジョブ内で複数個使用する時異った番号(3桁以内)を指定する。	番号はなにもつかない。
OUTPUT = REMOTE	実行時のシステム出力装置が端末となる。	センタのラインプリンタとなる。

C1) 利用者がカードあるいはファイルからコンパイルしたFORTRANでそれ以外にFORTRANライブラリしか使用しないジョブのみ

\$FORTRAN GOでやれる。効率がよく、CPU時間も少なくすむ。

⋮
\$GO

データ

② RUN

1	2 3 4	5	
	命 令		オ ペ ラ ン ド
\$	R U N		{EBNAME=実行形式プログラム名}{,FILENAME=専用ファイル名}{,LP=n}{,TIME=n}{,ERR=n}{,DATA=データ名}{,MACROSTEP=n}{,OUTPUT=REMOTE}

機能：このステップ前に作成した実行形式プログラムの実行、登録したEBファイルの実行形式プログラムの実行をする。

パラメータの説明

パラメータ	記 入 し た と き	省 略 し た と き
(注1) EBNAME= 実行形式プログラム名	指定した名前の実行形式プログラムの実行をする。	EXQTPRGMとなる。
C 1) FILENAME= 専用ファイル名	登録した専用ファイルの中でC 1)で指定した実行形式プログラムの実行をする。	これ以前のステップで作成した実行形式プログラムの実行をする。
LP = n	実行時の出力ページ数をn枚で打切る。nはそのジョブ種別の打切り以下でないといけない。	ジョブのトータルの出力が対応する種別の打切り定数で打切られる。
TIME = n	実行時のCPU時間をn秒で打切る。nはそのジョブ種別の打切り以下でないといけない。	ジョブのトータルのCPU時間が対応する種別の打切り定数で打切られる。
ERR = n	エラーの個数がnになったらジョブを打切る。	n = 50となる。
DATA= データ名	磁気テープ、大記憶にLIBEで作成したデータ名を指定する。 この時にはカード入力できない。	空となる。

パラメータ	記入したとき	省略したとき
MACROSTEP = n	このマクロをジョブの中で複数個指定する時異った番号（3桁）を与える。	番号はなにもつかない。
OUTPUT = REMOTE	実行時のシステム出力装置が端末となる。	センタのラインプリンタとなる。

(注1) L I E DのNAME文で与える名前でないといけない。

頭4文字がEXQTで始まる38文字以内の名前

C 1) 例) \$NO (EXQTIONという名前のプログラムを実行する。)

\$QJOB

\$RUN EBNAME=EXQTION, FILENAME=QU, EB. 00001

データ

\$JEND

6. 結合編集、実行のためのジョブ制御マクロ

L I E D R U N

1 2 3 4 5 6 7 8 9

	命 令		オ ペ ラ ン ド
S	L I E D R U N		[NOLIST][,LMAP][,EBNAME=実行形式プログラム名][,STARTPROGRM=実行開始プログラム名][,LP=n][,TIME=n][,ERR=n][,DATA=データ名][,MACROSTEP=n][,OUTPUT=REMOTE]

機能：この制御マクロ以前に作成された相対形式プログラムを結合編集し、さらに計算を行なう際に使用するデータを読み込んでプログラムの実行を行なう。

パラメータの説明：

パラメータ	記 入 し た と き	省 略 し た と き
N O L I S T	L I E D のリストもエラーメッセージも出さない。	リストとエラーメッセージを出す。
L M A P	作成された実行形式プログラムのコアマップの状態を出す。	コアマップを出さない。
E B N A M E = 実行形式プログラム名	指定の名前の実行形式プログラムを作成し実行する。	E X Q T P R G M となる。
S T A R T P R O G R A M = 実行開始 プログラム名	指定した名前のプログラム単位より実行を開始する。	P R G . M A I N F O R T R A N , A L G O L の主プログラムより実行される。
L P = n	実行時の出力ページ数を n 枚で打切る。n はその種別の打切り以下でないといけない。	ジョブのトータル出力が対応する種別の打切り定数で打切られる。
T I M E = n	実行時の CPU 時間を n 秒で打切る。n はその種別の打切り以下でないといけない。	ジョブのトータル CPU 時間が対応する種別の打切り定数で打切られる。
E R R = n	エラーの個数が n 個になったらジョブを打切る。	n = 50 となる。
D A T A = データ名	磁気テープ、大記憶に L I B E で作成したデータ名を指定する。 この時にはカード入力できない。	空となる。
M A C R O S T E P = n	\$ L I E D R U N の組合わせをジョブ中で複数個使用する時異った番号 (3 桁以内) を与える。	番号はつかない。
O U T P U T = R E M O T E	L I E D , 実行時のシステム出力装置が端末となる。	センタのプリンタとなる。

C1) ALGOLの結合編集のためのマクロである。

FORTRANでは $\left(\begin{array}{l} \text{FORTRANライブラリ} \\ \text{SSL} \\ \text{センターで登録したライブラリ} \end{array} \right)$ を使用するジョブを

結合編集実行のためのマクロである。

7. カード出力、紙テープ入力、プロッタを使用するためのジョブ制御マクロ

① SYSPCH

1	2 3 4 5 6 7	8
命 令		オ ペ ラ ン ド
\$	SYSPCH	〔OUTPUT=REMOTE〕

機能：出力としてカードパンチを出させる場合、リモートバッチでセンターと端末に出力を振り分ける時に使用する。カードパンチを出させる実行時のデータカードと次の制御文の間に入れる。データカードがない時は実行のための制御文と次の制御文の間に入れる。

パラメータ説明

パラメータ	記 入 し た と き	省 略 し た と き
OUTPUT =REMOTE	リモートバッチで WRITE (6,…)はセンターへ WRITE (7,…)は端末へ 出力する。 C1)	カードパンチの出力をする。

C1) リモートバッチで出力を振り分ける時に使用して下さい。

例) \$NO

ソースプログラム

\$LIEDRUN WRITE (6,…)はセンタのラインプリンタへ

データ

\$SYSPCH \$SYSPCH OUTPUT=REMOTE WRITE(7,…)は端末へ結果が
\$JEND である

(注) WRITE (6,)もWRITE (7,)両方REMOTEの指定はできない。

WRITE(6,)をREMOTEとした時にはSYSPCHは使えない。

WRITE(7,)によりモートバッチでカード出力は出せない。

② PTR

1	2 3 4	5
命 令		オ ペ ラ ン ド
\$	PTR	な し

機能：FORTRAN実行時のデータを紙テーブリーダより与える。

紙テーブリーダを使用する実行時のデータカードと次の制御文の間に入れる。データカードがない時は実行のための制御文と次の制御文の間に入れる。

例) \$NO

\$QJOB

ソースプログラム

\$LIEDRUN

データ

\$PTR

\$JEND

(注) これを使用するジョブは特殊ジョブである。

③ PLOTTER

1	2 3 4 5 6 7 8	9
命 令		オ ペ ラ ン ド
\$	PLOTTER	な し

機能：ALGOL, FORTRANでXYプロッタを使用する。XYプロッタを使用する実行時のデータカードと次の制御文の間に入れる。データカードがない時は実行のための制御文と次の制御文の間に入れる。

(注) これを使用するジョブは特殊ジョブである。

例) データカードがあり、\$SYSPCH, \$PTR, \$PLOTTERも入れる時

\$NO

\$QTOB

ソースプログラム

```
$ LIEDRUN
```

データ

```
$ SYSPCH
```

```
$ PTR
```

```
$ PLOTTER
```

```
$ JEND
```

8. FORTRAN, ALGOL中間書込みファイルの利用

F. WORK, A. WORK

1 2 3 4 5 6 7 8

	命 令		オ ペ ラ ン ド
\$	<言語の頭文字>・ WORK		{UNITNO=nn}{,RECORDSIZE=n}{,TRACK=n}

機能：FORTRAN,ALGOL実行時中間書き込みファイルを定義する。

ファイルを使用する実行時のデータカードと次の制御文の間に入れる。

データカードがない場合は実行のための制御文と次の制御文の間に入れる。

パラメータ	記 入 し た と き	省 略 し た と き
UNITNO = n n	使用する装置機番をゼロを省略しないで記入する。 ALGOL 01~17 但し17はGET, PUT用 FORTRAN 01~04 08~99	省略できない。
RECORDSIZE = n	1ブロックの大きさをバイト数で記す。 C 1)	1 1 5 2 バイト
TRACK = n	集団ディスクパックのトラック数を指定する。 C 2)	1 0 0 トラック

C 1) バイト数の計算

$$\text{語数} \times 9 / 2 \quad 256W \times 9 / 2 = 1152$$

C 2) 1152バイトの時には

1トラックに5ブロックはいる。

FORTRANでレコードのサイズをかえる場合

制御文のRECDSIZE=nで指定するとともにFORTRANのOPTION文でBUFFER=mを指定してください。

RECDSIZE=64×m×9/2としてmを計算します

省略するとm=4となります。

9. カード穿孔コード切り換え用制御文

1	2 3 4 5 6 7	8	
	命 令		オ ペ ラ ン ド
\$	CONV		な し
\$	NOCONV		な し

機能：CONV……この制御文に続くカードがIBM H型カード穿孔機で穿孔されていることを示す。

NOCONV……この制御文に続くカードがIBME L型カード穿孔機で穿孔されていることを示す。通常はNO文がこの機能を持つので不要である。

10. ジョブの終りを示す制御文

JEND

1	2 3 4 5	6	
	命 令		オ ペ ラ ン ド
\$	JEND		な し

標準的なジョブのカードデッキの構成法につきましては本広報の解説の欄に解説しております。