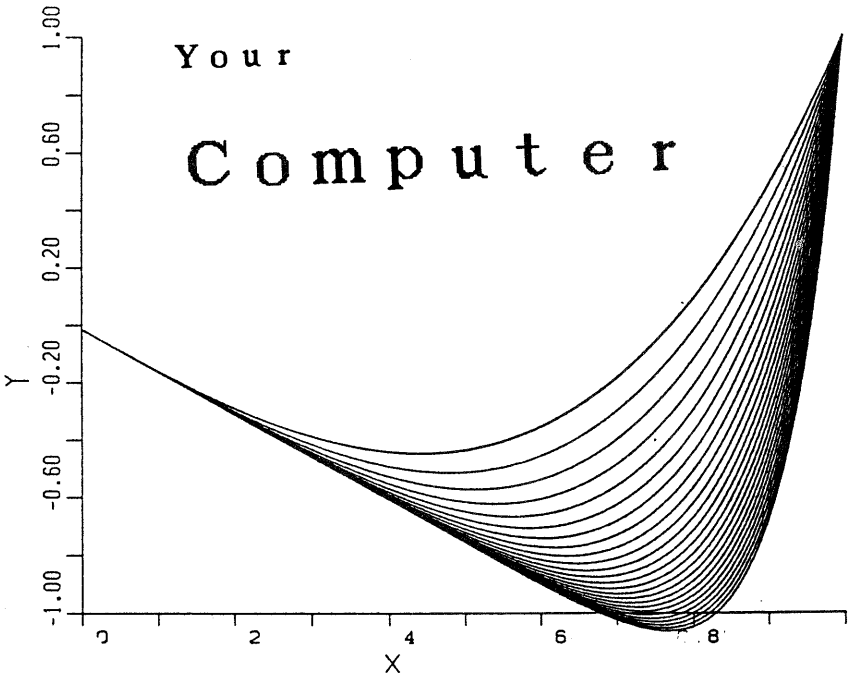
千葉大学情報処理センターニュース

昭和61年12月1日発行

第44号　　千葉大学情報処理センター



〔目　　次〕

——センター より——

1．仮想メモリーサイズの変更について

11月1日から仮想メモリーサイズの制限値を、下記のように変更致しました。

(1)バッチ

クラス	現行 リージョンサイズ		変更値 リージョンサイズ	
	標準	最大	標準	最大
A	250KB	1,344KB	250KB	3,072KB
B	512KB	1,344KB	1,024KB	3,072KB
C	512KB	1,344KB	1,024KB	3,072KB
D	1,024KB	3,008KB	3,072KB	5,120KB
J	256KB	1,344KB	1,024KB	3,072KB
K	1,024KB	3,008KB	3,072KB	5,120KB
H	192KB	256KB	640KB	1,024KB
M	480KB	1,024KB	1,024KB	3,072KB

(2)TSS

クラス	現行 リージョンサイズ		変更値 リージョンサイズ	
	標準	最大	標準	最大
教育用	320KB	512KB	640KB	1,024KB
研究用	512KB	1,024KB	1,024KB	3,072KB

2．SPSSからSPSS×への移行のお願い

社会科学用統計計算パッケージSPSSはSPSS×が公開されたことに伴い、12月からSPSS×のみの公開となりますので、利用者の皆様には、SPSS×を利用して下さるようお願い致します。

資料 1

—— 数式記号の清書出力 ——

センターでは、英文データを清書するプログラムであるRUNOFFのバージョンを上げました。これに伴い、機能も追加されました。それらの機能の概要は第40号(1986.4.1)で紹介したので参照して下さい。

今回は、清書出力を行うのに必要な制御語について述べます。制御語は英文データや数式データなどを文書処理した後、清書出力(漢字プリンター，H－８１７４)を行います。

１．数式データの編集

数式データの編集には次の２通りの方法があります。．MS，．MEと＃Mで囲む方法であります。数式データは数式記号などを記述します。

(1) ．MS と ．ME で囲む

```
.MS
数式データ
.ME
```

(2) ＃Mで囲む

＃M△数式データ△＃M

２．入力規則

(1) 大文字と小文字

入　力		出　力
A	→	A
a	→	a
&A	→	a
&a	→	A

㊟ギリシャ文字は￥か￥＆で入力すること。

(2) 区切り記号

数式編集指示子の区切記号は１個以上の空白(スペース)であります。空白を出力したいときは空白指示子(@Fn)を使用します。

㊟スペース文字以外の特殊文字も区切記号になる。これらについては、RUNOFF解説／文法マニュアルのP.42を参照して下さい。

(3) 添　字

添字はアポストロフィとコンマで指定します。

添字位置	出力形式	入力方法
右　上	X^{a}	X' a
真　上	$\overset{a}{X}$	X'' a
左　上	${}^{a}X$	X'''a
右　下	X_{a}	X, a
真　下	$\underset{a}{X}$	X,, a
左　下	${}_{a}X$	X,,,a

㊟添字をネストさせる場合は不等号記号(＜＞)で囲んで下さい。

$X^{a^{b+c}}$　　X' ＜a' b + c＞

(4) 分　数

分数の指定は分子と分母の間に斜線(／)を入れて下さい。

(5) 根　号

根号の指定はパーセント(%)であります。

3．数式編集機能

数式編集指示子は先頭が@記号で始まります。この指定子で書体、空白文字および数式の位置などが変更できます。

4．数式記号

数式記号の入力は使用例を参照して下さい。

5．使用例

(1) 数式とベキ乗

①．MSと．MEで囲む

```
.PO 5
Exampl 1
.SP 2
.MS
SIN'2 x+COS'2 x=1
.ME
```

Exampl 1

$\sin^2 x+COS^2 x=1$

②＃Mで囲む

```
.PO 5
Exampl 2
.SP 2
#M SIN'2 x+COS'2 x=1 #M
```

Exampl 2

$\sin^2 x+COS^2 x=1$

(2) 総和記号

```
.PO 5
Exampl 3
.SP 2
#M SUM,,i=1''n X,i #M
```

Exampl 3

$\sum_{i=1}^{n} X_i$

(3) 添 字

```
.PO 5
Exampl 4
.SP 2
#M A,ij =<(B,ij +C,ij )>/D,ij#M
```

Exampl 4

$A_{ij}=\frac{(B_{ij}+C_{ij})}{D_{ij}}$

(4) 根 号

```
.PO 5
Exampl 5
.SP 2
#M %<1/a +b> #M
```

Exampl 5

$\sqrt{\frac{1}{a}+b}$

(5) 立ち根号

```
.PO 5
Exampl 6
.SP 2
#M %'3 <1/a +b> #M
```

Exampl 6

$\sqrt[3]{\frac{1}{a}+b}$

(6) 積分記号

```
.PO 5
Exampl 7
.SP 2
#M Y,1 = INTEGRAL,a'b (x+1)dx #m
```

Exampl 7

$Y_1=\int_a^b (x+1)dx$

(7) 書 体

①イタリック体

```
.LS 1
.FO ''-%-''
.PL 30
.SY I
.HE ''Operating Sytem of HITAC M/L Series Computers''
INTRODUCTION

As operating systems(Os) of HITAC M series
computers (hereinafter referred to simply as the M series),
Virtual Storage Operating System(VOS) 1,VOS2 and
VOS3 were developed. Similarly,VOSO and VOS1-S were
developed for the HITAC L series computers (L series).
For these operating systems,consistent design targets
```

```
                    Operating Sytem of HITAC M/L Series Computers
INTRODUCTION

As operating systems(Os) of HITAC M series computers (hereinafter referred to
simply as the M series), Virtual Storage Operating System(VOS) 1,VOS2 and
VOS3 were developed.  Similarly,VOSO and VOS1-S were developed for the HITAC
L series computers (L series).  For these operating systems,consistent design
```

②ローマン体

```
.LS 1
.FO ''-%-''
.PL 30
.HE ''Operating Sytem of HITAC M/L Series Computers''
INTRODUCTION
.SY R
As operating systems(Os) of HITAC M series
computers (hereinafter referred to simply as the M series),
Virtual Storage Operating System(VOS) 1,VOS2 and
VOS3 were developed. Similarly,VOSO and VOS1-S were
developed for the HITAC L series computers (L series).
For these operating systems,consistent design targets
```

```
                    Operating Sytem of HITAC M/L Series Computers
INTRODUCTION

As operating systems(Os) of HITAC M series computers (hereinafter referred to
simply as the M series),  Virtual  Storage  Operating  System(VOS) 1,VOS2 and
VOS3 were developed.    Similarly,VOSO and VOS1-S were developed for the HITAC
L series computers (L series).  For these operating systems,consistent design
```

(8) ギリシャ文字

```
.NF
.MS
"¥&A"@F1"¥&B"@F1"¥&G"@F1"¥&D"@F1"¥&E"@F1"¥&Z"@F1"¥&H"@F1
.ME
.MS
"¥&C"@F1"¥&I"@F1"¥&K"@F1"¥&L"@F1"¥&M"@F1"¥&N"@F1"¥&X"@F1
.ME
.MS
"¥&O"@F1"¥&P"@F1"¥&R"@F1"¥&S"@F1"¥&T"@F1"¥&U"@F1"¥&F"@F1
.ME
.MS
"¥&Y"@F1"¥&Y"@F1"¥&W"
.ME
```

$\alpha\ \beta\ \gamma\ \delta\ \varepsilon\ \zeta\ \eta$
$\theta\ \iota\ \kappa\ \lambda\ \mu\ \nu\ \xi$
$o\ \pi\ \rho\ \sigma\ \tau\ \upsilon\ \varphi$
$\chi\ \phi\ \omega$

(9) 二段組み

```
.FO ''-%-''
.LL 50
.SS
.HE ''Operating Sytem of HITAC M/L Series Computers''
INTRODUCTION
.MC 0 25 21
As operating systems(Os)
of HITAC M series
computers (hereinafter
referred to simpiy as the
M series), Virtual Storag
e Operating System (VOS)1
,VOS2 and VOS3 were devel
oped. Similarly,VOSO and
VOS1-3 were developed for
the HITAC L series comput
ers(L series).
For these operating syste
ms,consistent design
targets
```

```
 Operating Sytem of HITAC M/L Series Computers


INTRODUCTION

As          operating    oped.  Similarly,VOSO
systems(Os)  of HITAC    and    VOS1-3    were
M  series   computers    developed   for   the
(hereinafter referred    HITAC L series comput
to  simpiy  as  the M    ers(L  series).   For
series).       Virtual    these operating syste
Storag  e   Operating    ms,consistent  design
System  (VOS)1  ,VOS2    targets
and VOS3  were  devel
```

6. 実　行

```
READY
FOUT @.DATA A4V ROFF
*** ROFF NORMAL END ***
READY
```

@. DATA : RUNOFFの制御語及び英文データが記録されているデータセット名。
A4 V　　: A 4 の大きさで縦方向に出力する。
ROFF　　: 英文清書出力

㊟FOUT コマンドプロシジャーのパラメータの詳細はセンターニュースの第40号を参照して下さい。

〔参考文献〕

1. VOS3 RUNOFF 解説/文法書 (8090 - 3 - 312 - 20) ：日立マニュアル

資料 2

―― 最適化FORTRAN77での上手なデバッグ ――

1. はじめに

FORTRAN言語（この節では最適化 FORTRAN 77のこと。）のプログラムを作成するとき、もっとも時間的、精神的なロスと感じるのはデバッグのときである。これは次の原因によっているからである。

① 誤りのないプログラムの実行速度が最も速くなるような標準コンパイルオプションとなっている。従って、どこでどの様なエラーが発生したかプログラマはわからない。

② システムからのメッセージも上記の目的をかなえるために極力抑えている。

以上のことを考えると、プログラム作成、実行時には逆に

① 誤りのあるプログラムとして、実行速度は犠牲にしても、どこで、どの様なエラーが発生したかを表示させるコンパイオプションとする。

② システムからのメッセージをなるべく表示させて理由を知るために役立てる。

ということが重要になってくる。2.3.ではその方法と実例を挙げる。このことは現在既に作成されているプログラムについても行えば、意外な誤りが発見される場合もあるであろう。

2. 環境設定

① システムからのメッセージをなるべく表示させるためには次の様にする。

(イ) ＴＳＳの場合

READYモードから以下のアンダーラインのコマンドを投入する。

```
READY
prof pause msgid prompt wtpmsg recover
```

オペランドの説明は「プログラムプロダクトVOS3/SP21 TSSコマンド」を参照されたい。

(ロ) BATCHの場合

ＪＣＬのジョブ(JOB) 文以下のMSGCLASSオペランドを追加する。

```
//利用者番号  JOB,
//MSGCLASS=(1,1), ……
```

以上の操作は FORTRANに限らず他の言語等でも有効です。

② プログラム作成上の心がけ

(イ) デバッグ、管理しやすいプログラム

◇見やすいコーデングにしておくことは、見直すときにとても重要なことである。

◇具体的にはネストしている場合は５カラム程度下げて記述する。

◇GOTO文は極力避ける。

◇自由形式だからといって先頭カラムからベタ書しない。

（特に継続行は - を使うために初心者は演算記号と間違えやすいので固定形式をすすめる。ライブラリー、バッチ、DESPでは標準が固定形式となっている。固定形式で揃えていたほうが当然よい。）

◇必ずプログラムの作成日、機能、変数の説明をいれる。

◇できるだけプログラムをモジュール（部品）化しておく。例えば、入力のサブルーチンとか作表のサブルーチンとかいうようにしておく。この部品を今後のプログラム作成時に利用すれば、この部分のデバッグから免れることができる。

(10000行に及ぶFORTRANプログラムを持ってこられても相談員がすぐに対応できない。)

例1 (悪い例)

```
1 dimension a(100)
10 read(5,1020) (a(i),i=1,100)
11 PI=0.234
12 1020 format(20F10.5)
100 do 1200 i=1,30,2
101 do 1200 j=1,12
105 if (i.LE.10) then 1200
120 a(j+i)=sin(float(i+j))/pi+cos(float(i)) -
121 **2*(tan(float(i))
130 1200 continue
140 stop
151 END
```

例2 (例1を固定形式で書き換えたもの)

```
1000 C                                                    ←プログラムのドキュメントを
1010 C   program : suuchi keisann no tameno TABLE sakusei          必ず入れる
1020 C     var A : = hairetu  ookisa      1...100 : REAL type
1030 C   create  :     1986/11/12
1040 C   mod.     :     1986/11/16  lien 1200
1050 C   reff.    :   J.R.S.A(1865) Vol. 24, NO. 4, p.p. 413
1051 C
1052         PARAMETER ( PI=0.234 )                    ←定数はPARAMETER文
1060         REAL A
1070         DIMENSION A(100)
1080 C
1090         READ(5, '(20F10.5)' )  ( A(I),I=1,100 )←FORMAT文が多い場合
1100 C                                                  このようにすると紛れない
1110
1110         DO 1000 I = 1, 30, 2
1120            DO 1100 J = 1, 12                      ←異なるDO文に対して
1140               A(J+I) = SIN( FLOAT(I+J))/PI +          同じ文番号を使わない
1150      &                    COS( FLOAT(I) )**2*(TAN( FLOAT(I) )
1160  1100      CONTINUE
1170  1000   CONTINUE
1180 C
1190         STOP
1195         END
```

③デバッグ用コンパイルオプション

(イ)＊PROCESS文を活用する。

前記　例2では行番号1000を以下のように変更する。

```
1000*PROCESS FIXED SUBCHK ARGCHK NUM DLINE FIXEOVER RUNOPT OPT(0)
```

意味を下記に示す。

FIXED	固定形式である。
SUBCHK	配列等で添え字の範囲をチェックする。
ARGCHK	関数、サブルーチン等の引数の不一致をチェックする。
NUM	エラーの場合ISN番号でなく行番号で示す。
DLINE	デバック行として？を使用する。 先頭１カラム目の？は空白とみなす。NODLINE指定時は注釈行となる。
FIXEOVER	固定小数点オバーフローの検出をおこなう。
RUNOPT	実行時オプション指定可能とする。

これによって例３の様にかなり親切にエラーを表示可能となる。

例３

```
READY
EDIT SAMPLE.FORT F7(F1)        ←
EDIT
LIST
10 *PROCESS FIXED SUBCHK ARGCHK NUM DLINE FIXEOVER RUNOPT OPT(0)
15        PROGRAM MAIN
20        DIMENSION A(100)
30 C
40         DO 30 I=1,101
50            READ(10,*) A(I)
60     30  CONTINUE
70    ....

90        END
EDIT
V                    ←EDITにはいったら必ず行う。
SCAN                 ←1行内で文法の誤りがあるか調べる。
90        END        ←誤りがない場合は最終行が表示される。
RUN
*OFORT77 ENTERED
JML1941 SUBCHK - SUBSCRIPT VALUE OUT OF RANGE  ARRAY NAME A --
              SUBSCRIPT VALUE    ---   '0000101'
                                         ↑   添字の値の範囲を越えた配列名称
                                    その時の添字の値
         TRACEBACK  ROUTINE  ISN  ......
                    F#SCHK
                    MAIN     00000030
                      ↑          ↑
                現象の起こったルーチン名   その行番号
```

３．エラーが起こった場合の処置

①JMKXXXのエラー

文法上のエラー(具体的にはFORTRANコンパイラーから)

プログラムプロダクト　VOS２，VOS３

最適化FORTRAN７７、FORTRAN HAP使用の手引(8080－３－258－70)付録B.を見れば良い。

②JMU／JMNのエラー

プログラムプロダクト　　VOS３

最適化FORTRAN７７、端末使用の手引(8080―３―222―10)付録A.を見れば良い。但し、JMU のエラーで手元にマニュアルが無いときは①でのマニュアルのJMKの同じエラー番号を見れば多くの場合ことたりる。

③JML２４０のエラー

例４のように××××のシステムリターンコードが示される。

例４

JML240I　ABNORMAL　TERMINATION－ABEND　CODE＝XXXX　USER　YYYY……

XXXXがOC1、OC2、OC3、OC4、の場合の理由は次の通り。

配列の添字範囲を越えている。

サブルーチンの引き数、型の不整合。

サブルーチンを呼ぶ側は定数であるが、帰る側でそこに値を入れている。

これらは例３の様な＊PROCESS文を挿入するとエラー箇所がわかる。

YYYYがOOOOの場合はGPSL等実行時に起こる。これはVSINT等の初期設定が不正である理由が考えられる。

④JBBXXXXのエラー

基本的にデータの入出力のエラーで、DD名称、データセット名称、リターンコードが現れる。データセット名称がユーザーの場合はデータセットの形式、大きさが不正であることを示している。特によく起こるのはリターンコードがSB３７で大きさが不足していることを示している。データセット名称の先頭１文字が@の短期データセットにして初期値、増分値をシリンダー割当で行ってみると良い。

その他DD名称がSYSOOXXXであり、データセット名称に自分で覚えが無い場合は作業用データセットである。バッチのFORT７CLGのカタログプロシジャーで行ってみて、不足する作業領域を明示的に大きくするとよい。

⑤システムリターンコードが１０６、８０Aの場合

仮想記憶の不足、不正である。TSS時は以下のように再LOGONするとよい。

LOGON利用者番号／パスワード　S(３０７２)

４．その他エラーの調べ方

①VOS３システムメッセージ／システムコード8090―９―703の「はじめに」の表でエラーメッセージがどのマニュアルにあるか調べる。

②①で示されたマニュアルの付録のメッセージを調べる。参考、参照マニュアルが記載されていればそのマニュアルを調べる。

以上の結果で不明な場合は平日10:00～3:00(昼休み、土曜を除く）の時間帯にプログラム相談員に聞く。更に不明な場合はセンター西條、橋本まで。

５．最後に

「エラーメッセージが出力されないからといって、プログラムにエラーが存在しないという保証は無い。」ということにご注意されたい。

■発　行／千葉大学情報処理センター　千葉市弥生町１―33　TEL.0472(51)1111・内　線3440・3441
■印刷所／株式会社 宣 巧 社　　千葉市朝日ヶ丘町2492-8　TEL.0472(58)3991　FAX.(58)3439