[05_01]九州大学大型計算機センター広報表紙奥付等

出版情報：九州大学大型計算機センター広報. 5 (1), 1972-02-22. 九州大学大型計算機センター
バージョン：
権利関係：

利 用 者 の 声

1．文法書によって作られた「もの」の解説記事について(九州大学 工学部 応用原子核 吉岡啓介)

これから数回にわたって解説される御予定の内容には、非常な期待を寄せている者の一人であるが、以下に述べる事項にも言及して戴くことを希望します。

イ） 現行の文法書、解説書で言及していないことについて

やってはならない場合と、チェックを受けないのでやってもかまわない（利用者の希望した動作はやってくれる）場合とがある。

例1． VARIABLE FORMAT で倍精度単純変数には書式を入れることは出来ないのではないか。

例2． ADJUSTABLE ARRAYで2次元の場合、第2添字は実際に宣言されている値よりも小さくてかまわない。利用者の方で陽に受け渡しをしなかった残余の部分が内容の変更を受けるか、受けないかによってこの性質が利用出来るかどうかの判断の分かれになる。

例3． レベル0の閉カッコの直前に“,”があってはならない。

例4． DOループの制御変数の値によって、ループ内のある変数の値が定められるとき、ループ内を通る条件判定が通る前になされるのか、通った後かということは利用者の方で注意していなくてもよいか。

例5． サブルーチンの引用において、EXTERNAL文で引用が指定されている場合、実名で他のサブルーチンから引用される場合もふくめて、常にサブルーチンの排列順序は問題にしなくてもよいのか。

（回答） 上記の要望について内容を検討した結果、広報の解説のページに掲載することは、省略させて戴くことになりました。ここで簡単に説明するということで御了承願います。

例1については、FORTRAN 解説編II（EX－061－3－4）P.53を参照してください。書式が与えられるのは配列に限られ、型は整数型、実数型、倍精度実数型、論理型のいずれであってもよい。書式はDATA 文、READ文で与えなければなりません。

例2については、FORTRAN 解説編II（EX－061－3－4）P.68を参照してください。2次元にかかわらず添字は、呼び出しプログラムの側で宣言されたものより小さくてかまいませんが、各配列の要素に対応関係のずれが生じてきます。2次元において、第2添字のみ小さい場合は対応関係のずれはありません。また、利用者が受け渡しをしなかった部分の内容がこわれることはありません。

〔例〕

整合配列を使った副プログラムをCALLするプログラム

```
REAL  X(3,3)
.....
CALL  SUB(X,2,3)
.....
CALL  SUB(X,3,2)
.....
```

最初のCALL文では

X(1,1) ……A(1,1)
X(2,1) ……A(2,1)
X(3,1) ……A(1,2)
X(1,2) ……A(2,2)
X(2,2) ……A(1,3)
X(3,2) ……A(2,3)
X(1,3)
X(2,3)
X(3,3)　} 内容がこわされることはない。

の対応となる。

整合配列を使った副プログラム

```
SUBROUTLNE  SUB(A,M,N)
REAL  A(M,N)
.....
.....
.....
.....
```

2番目のCALL文では

X(1,1) ……A(1,1)
X(2,1) ……A(2,1)
X(3,1) ……A(3,1)
X(1,2) ……A(1,2)
X(2,2) ……A(2,2)
X(3,2) ……A(3,2)
X(1,3)
X(2,3)
X(3,3)　} 内容がこわされることはない。

の対応となる。

例3については、テストの結果“,”があっても、正常な結果が得られました。

例4については、FORTRAN文法編（SP-061-4-4）P.40を参照してください。

制御変数の条件判定はDOの範囲を実行する前になされています。DOが満足された場合は、制御変数は不定となり、満足されずにDOループを飛び出した場合は、その直前に与えられた値をもっています。

例5については、カードデックの構成順序についての疑問だと思われます。サブルーチンが実引数にあらわれた場合と、CALL文で引用された場合とにかかわらず、呼び出されたサブルーチンは、呼び出しプログラムより前にあっても、後にあっても問題はありません。

ロ）optimize について

1　どのような点をoptimize しているのか。および利用者がコーディングするとき、どのような点に留意すればoptimize したことと同等になるか。

（回答）　optimize の主なものには次の3項目があります。

(1) 共通式の消去……同じ計算結果をもたらす式の一括化

```
X = A + B                    T = A + B
                    ——→      X = T
Y = A + B + C                 ⋮
                              Y = T + C
```

※　実数値をもつ式の場合にはTの代入の際にレジスタにある値を丸めて格納するので、CタイプとDタイプとで異なった結果になることもあります。

(2) 配置転換……繰返し文中の計算で、繰返しの外に出せるものは出す。

```
      DO  1  I = 1, 10                T = A + B
      X (I) = A + B        ——→        DO  1  I = 1, 10
1     CONTINUE                        X (I) = T
                                    1   CONTINUE
```

※　繰返しの中にIF文がある場合に、プログラム上の論理は正しくても、optimize したために繰返しの外に出された計算で、オーバーフロー、デバイドチェックなどのエラーとなることがあります。現在、optimize されるのとされないのとがありますので、その例をあげてみましょう。

・optimizeされる例

```
      DIMENSION   A(10)
      DO    1  I=1,  10
      DO    1  J=1,   5
      IF(I.LT.2 * J) X = A(I+1) - A(I)
1     CONTINUE
                  ↓
      DIMENSION  A(10)
      DO  1  I=1,  10
      T = A(I+1) - A (I)
      DO  1  J=1,  5
      IF (I.LT. 2 * J)  X = T
1     CONTINUE
```

このようにoptimizeした結果、I＝10のときA(I＋1) は宣言の範囲を起えてしまい、エラーの原因となることがあります。

• optimizeされない例（以前はされていたが、現在はされなくなった）

上と同じ計算ですが、次の場合にはされません。

```
      DIMENSION A(10)
      DO  1 I=1,10
      DO  1 J=1,5
      IF(I.GE. 2*J) GO  TO  1
      X=A(I+1) -A(I)  ……①
    1 CONTINUE
```

この例では、①の式はJのLOOPとは無関係なので

```
      DO   1  I=1, 10
      IF(I .LT. 10) T=A(I+1) -A(I)  ……②
      DO   2  J=1, 5
      IF(I .GE. 2) GO  TO  1
      X=T
    1 CONTINUE
```

とプログラミングすると良いでしょう。この場合、②の判定文がないとI＝１０のとき　A（I＋1）は宣言の範囲を越えてしまい、エラーの原因となることがあります。

```
      DO  1 I=1, N
      A(I) =B(I)
      IF(X.GT. 0. 0) A(I) =SQRT(X) + B(I)
    1 CONTINUE
```

これは、次のようにプログラミングすればよいでしょう

```
      T= 0. 0
      IF(X.GT. 0. 0)T=SQRT(X)
      DO  1 I=1, N
      A(I) =T+B(I)
    1 CONTINUE
```

(3)　命令の置換……計算時間が短い命令に置換える。

番地計算等のレジスタを効果的に使う。

レジスタの効果的な利用については、プログラマの関与できない部分ではありますが、最もoptimizeの効果があらわれる部分です。

判定文において、論理演算子（.AND.,.OR.,.NOT.）を使用するとoptimizeされません。

```
      IF(A.EQ. B.OR. C.NE. D) GO  TO  10
```

次のようにすると　optimize されて、命令ステップ数が半分以下になります。

↓

```
      IF (A.EQ. B) GO  TO  10
      IF (C.NE. D) GO  TO  10
```

なお、以前は定数および定数を暗に含む計算はコンパイル時に行なっていましたが、現在は、行なっていません。

以上まとめると

- 同一計算はできるだけ一括し、むだな繰返しをはぶく。（特に(2)で述べたような考慮は配ったほうがよい。）
- 論理ＩＦより算術ＩＦの方が早い
- 配列をふんだんに使用している計算は、optimize させるとレジスタを有効に使うので番地計算時間が短縮される。

ということがいえるでしょう。

2 多重ループの端末文を一致せしめたときとていねいにループごとに分けたときとで所要メモリ一数に変化があるか

（回答） ＣタイプとＤタイプとの比較では、Ｄタイプがステップ数が少し短くなりますが、それぞれのタイプ内では、端末文を一致してもしなくても変わりません。

3 DO LOOP の繰返し終端値を算術式Ｍ＋Ｎで表示するとき Ｋ（＝Ｍ＋Ｎ）で表示するときとでLOOP の繰返し条件を判定するための所要時間の比率はどの程度か

（回答） DO 1 I=1, M+N の場合でもＭ＋Ｎをあらかじめ計算していますので違いはほとんどありません。

4 論理ＩＦで．LE. と．LT. との判定条件の比率はどの程度か

（回答） ．LE. はequal 判定を行ないますので ．ＬＴ．よりその分だけ２ステップ多くなっています。

判定回数が少ない場合はほとんど差はありませんが、繰返しが重なった場合、次のようになりました。

		Ｃタイプ	Ｄタイプ（単位ミリセカンド）
１００，０００回	LT	４１４６	２８２６
	LE	４３２０	２９８６
２００，０００回	LT	８２６１	５８５９
	LE	８８９２	５９７０

ハ)　エラーチェックについて

i　$FORTRAN, $LIED, $RUN の各STEPで検出される文法的エラーはどの程度までか。

（回答）　$FORTRANではFORTRAN ステートメントの記述形式の誤まり、未定義文番号、DO の入れ子の誤まり、プログラムの構成要素と構成順序の誤まりなどの検出を行ないます。

READ文やWRITE における入出力並びと対応する欄記述子の型のチェック、関数や副プログラムの引用における実引数と仮引数の順序、個数および型のチェックなどは行ないません。

$LIEDではLIED 制御文の記述形式の誤まりなどの検出を行なう他、実行形式プログラムの大きさが$QTOBで指定したCOESIZE を越えた場合、ＬＩＥＤの作業領域が足りない場合、未定義記号がある場合などはその表示をします。

$RUNでは文やWRITE 文における入出力並びと対応する欄記述子の型が異なる場合、READ文やWRITE 文における入出力並びと対応する欄記述子の型が異なる場合、関数や副プログラムの引用における実引数と仮引数の個数が異なる場合などはその表示をします。

ii　サブルーチン副プログラムのパラメータ個数が、CALLされる場合に応じて自由に増減されるプログラムを見かけるが、どのような工夫がしてあるのか。

（回答）　FORTRANのOPTION文にOMIT機能があります。OMIT 機能は基本外部関数の引数の型のチェックを行なわない。副プログラムの仮引数と実引数の個数のチェックを行なわない。よって実行時間が短かくなります。

この機能は１９７１年８月１日より、FORTRAN レベルアップで追加されたものです。

パラメータの対応については、解説「FORTRAN を基礎にしたプログラム言語について（Ｉ）」（広報vol.3 No.5 P22以下の記述）を参考にして下さい。

2．現用機種の拡充、改良について （九州大学 工学部 応用原子核 吉岡啓介）

イ）ファイル開放量の増加

本年度は9月末で共用ボリューム私用ファイルは余裕が無くなってしまったが、ファイル利用希望者に対する充足率は非常に低い値であることが問題である。ファイル開放量の増加について一層の御努力をお願いしたい。ここで、貧しい利用者を締め出すようなことになる負担金増額については慎重な配慮をお願いします。

（回答） センターニュース No.27 でもお知らせしましたように、これまでＴＳＳ専用に使用しておりましたE.042を共用ボリュームとして解放することにいたしますので、10月に中止した新規の開設申請の受付を1月17日から再開しました。できるだけ多くの利用希望者に開放するために1課題で開設できるトラック数を40トラックまでに制限しております。
ファイルの使用負担金は現行のままです。

ロ）カーブプロッタについて

現在用いられているインクの色はみるからに汚ならしい。濃い紫色系統の色に換えられないか。なお現在のD job 操作で処理するなら、利用者の希望によってグラフ用紙は無地白色を使うことが出来るようにして頂けないか（必要な座標軸、目盛は利用者が画く）。またPENUPでの移動時のインクカスレが非常に目につく。機械的な改良をお願いしたい。

（回答） X-Y プロッタ で利用者の方々にご不便をおかけしていると思います。Ｘ－Ｙプロッタについての要望、苦情は他にもありましたので併せて回答いたします。

1．ボールペンについて

X-Yプロッタ用ボールペンとして現在赤、黒、緑、青の4色があります。それでセンターではPENUPでの移動時のインクのかすれ、用紙のよごれ、見やすさに重点をおいて幾度かテストし、一番良かった赤を使用しています。しかしながら、赤色は複写機ではコピーできない（ゼロックスでは可）ので困るという苦情が度々ありますので、検討のうえ、複写の確実な墨色ボールペンの改良を現在メーカー（岩崎通信機）に申し入れています。

2．用紙について

Ｘ－Ｙプロッタ用用紙としては無地のものと、グラフ用紙のものの二種類があります。現在当センターで使用しているのはグラフ用紙のものですか、無地を希望される方が多いので近々無地のものにかえる予定です。なお、利用者の希望にあわせて用紙を無地のものも、グラフ用紙のものも使用出来るようにすることは、ジョブ処理上（プロッタＪＯＢはＭＴ－OUTにする）できません。

ハ）ＴＳＳ時間帯の増加

現在の機種構成では将来ともＴＳＳ処理とローカルバッチ処理とを並行させて走らせることは無理な状況であろうか。少なくとも毎日２～３時間のＴＳＳ処理が出来るような工夫は考えられないか。

ＴＳＳ用ファイルの不足も目立っているのであるが、（端末機１台について同一時期のファイル利用者数を10人平均とすれば30台で 15000トラック入用）センター職員の方々の御努力によって、ローカルバッチ用ファイルが直接利用できるようになれば１課題についての許容トラック数は現行の50トラックから20トラック弱に縮少しても利用者の側からの不満の増加はそんなに増えないと思われる。

以上現在の機種構成では能力外かも知れないと思われる要求もふくめてお願いしましたが、現在の能力にまで計算機を開発して頂いたセンター職員の方々の御努力には（時間外運転という御苦労もふくめて）深く感謝しています。

（回答）　現在のところＴＳＳの時間帯を増加する予定はありませんが、本年１月からバッチ、ＴＳＳ両方のモニタのレベルが改良されましたので、ファイルは、ローカルバッチ、ＴＳＳの区別なく利用できるようになります。共用ボリュームE.042を利用するときには機番指定（UNIT＝E. 042）、ボリューム通番指定（VOLNO＝E00042）が必要です。

ＴＳＳ用のマクロのうち、$GETFILE, $DELFILE, $COPYFILE はなくなります。

3．ロビーに設置のＪＰ1500でのジョブ処理状況表示について

センターロビーにあるＪP1500（オンライン・タイプライター　Ｆ１５９２Ａ）から出る個々のジョブの入力・出力終了等のメッセージの読み方の説明書を側において欲しい（無記名）

（回答）　４６年１１月末からセンター二階ロビーのＪＰ１５００タイプライタを使用してジョブ処理状況を表示していますが、利用者の方むけのメッセージの説明書を用意していなかった為、メッセージが理解できなかった方もおられると思います。

メッセージの意味は本号68頁に記載しておりますのでご覧ください。

また、表示用タイプライターにも説明書を添付しています。

4．ジョブのターンアラウンド・タイムについて

以前はＡジョブで１時間以内に戻っていたが、最近はＡジョブで１日かかる。少なくとも、Ｘジョブで１時間以内、Ａジョブで３～４時間以内で戻して欲しい。（無記名）

（回答）　計算依頼におけるジョブ種別ごとのターンアラウンド・タイムはおおよそ下記の通りですが、10月～３月においては、計算依頼が集中し、普通時よりターンアラウンド・タイムが大幅にずれて来る事があります。センターとしましては、夜間にジョブ処理を行なう等、できるだけ利用者の方のご希望に添うよう努力しております。

Ｘジョブ	３０分～３時間	Ｃジョブ	１日以上
Ａジョブ	１時間～半　日	Ｄジョブ	ＪＯＢにより異なる
Ｂジョブ	半　日～１　日		