

[09_02]九州大学大型計算機センター広報表紙奥付等

<https://hdl.handle.net/2324/1468048>

出版情報：九州大学大型計算機センター広報. 9 (2), 1976-06-01. 九州大学大型計算機センター
バージョン：
権利関係：



昭和50年度後期プログラム相談票のまとめ

当センターでは、プログラム相談時に「プログラム相談票」への記入を行なっていただくようになっております。そこで昭和50年度後期分(50年10月～51年3月)のプログラム相談票をまとめてみました。

質問事項

利用手続き, 資料	1
コントロールカード	14
プログラムの書き方	9
エラーの見方, みつけ方	86
ライブラリプログラム	8
CPS	10
ファイル	10
XYプロッタ	13
磁気テープ	5
その他	1
計	157

エラーの原因

コントロールカード誤り	21
FORMAT文誤り	1
入力データ誤り	9
配列オーバー	6
COMMON変数の対応誤り	1
未定義変数使用	3
宣言文誤り	1
引数誤り	6
論理誤り	10
文法誤り	15
ライブラリ使用方法の誤り	6
システムエラー	5
原因不明	20
その他	53
計	157

使用言語

FORTRAN-D	120
FORTRAN-H	13
ALGOL	8
その他	16
計	157

エラーの現象

コントロールカードエラー	15
コンパイルエラー	16
結合編集(LIED)エラー	10
実行時のエラー	69
エラーメッセージはなく結果がおかしい	10
その他	37
計	157

相談結果

解決	88
利用者がチェック	36
未決	18
キャンセル処理	2
その他	13
計	157

- FORTRANHについて

FORTRAN-Hの利用者はまだまだ少ないようです。FORTRAN-Dよりも機能が拡張されており、他の機種との互換性もあります。（たとえば、FACOM230-28）

現在、FORTRAN-HコンパイラはE004(V02, L03)のものを使っていますが、これは当初のもの(E003)に比べて、かなり内容が改善されています。(OPTIMIZATION, コンパイラの大きさが43Kから41Kになっているなど)

現在、複素数型の配列要素名を、関数やサブルーチンの実引数として使用すると、コンパイル中にFT599Z INTERNAL COMPILER ERROR IDENTIFICATION NO. () というエラーメッセージが出て、異常終了します。(OPT0, OPT2共) これは、複素数型配列要素名を、複素数型単純変数に置きかえて使用することで避けられます。

FORTRAN-Hの標準関数の引用で、数人の利用者が同じエラーを起こしているのがありました。

4倍精度の引数が許されていない関数(双曲正弦, 逆正弦等)を呼び出して、LIED時にLD213X MITEIGI KIGOU ** F*QSINH ** のエラーを出していました。いずれも4倍精度の関数を直接引用しているのではなく、DOUBLEパラメータを指定したため、あるいは総称関数名(ASIN, SINH等)を引用したために起きたというものでした。

なお、総称関数名のASIN, ACOSを4倍精度の引数で引用すると、LD211Z CALL-BUN NO FILE GA LIBRARY NO KEISHIKI DE NAI ** FORTLIBH ** というエラーメッセージが出ますが、これは前記のLD213Xのエラーメッセージが出るべきものです。これについては、エラーメッセージが適切でないので、富士通に修正依頼をしています。

- 実行時のファイル利用について

最近、実行時のファイル利用によるファイルのアクセス回数が問題になっています。大量のデータをファイルに入出力すると、当然アクセス回数は多くなってきますが、できるだけアクセス回数を減らす対策として、ブロックを大きくする方法があります。現在、ブロック長は何も指定しないと、256語ですが、これは1600語まで大きくすることができます。(FORTRAN-Dの場合)

また、FORTRAN-Dではブロック長を大きくすると、OPTION文のBUFFER=nの指定をしなければなりません。

- OPTION文の使い方について(FORTRAN-D)

OPTION文は、次のOPTION文が来ると前のOPTION文の機能はなくなってしまいます。不用意にもOPTION文を何枚も入れたため、BUFFER=nのパラメータが働かなくて、実行時にFT732Z ERROR AT () *BLOCK SIZE>BUFFER SIZE * Fnnのエラーが起きたというようなことがありました。

また、OPTION文はパラメータの種類によって、その有効範囲が変わってきます。

以下にそれを示します。

OPTION 文の有効範囲

1. OPTION 文の次のステートメントより有効になるもの
 - LIST/NOLIST
 - SEQ/NOSEQ
 - X
 - DOUBLE (定数変換のみ)
 - NODEBUG
2. OPTION 文の2つ目以降のステートメントより有効になるもの
 - ASTER (または*)注
3. エレメントの最後のOPTION 文の指定が有効になるもの
 - MAP/NOMAP
 - FASP
 - DOUBLE (定数変換を除く)
 - OMIT
4. 翻訳単位の最後のOPTION 文の指定が有効になるもの
 - BUFFER=n

注 ASTER (または*)の指定は、すぐ後に続く文が、デバッグ行であるとき、それに続く全てのデバッグ行に対して無効となります。対策として、OPTION 文のすぐ後にCONTINUE 文を入れるとよいでしょう。

例

```

      OPTION ASTER
*      WRITE(6,100)A
*      WRITE(6,100)B
* 100  FORMAT(F10.5)
      DO 1  I=1,N
          :
      1  CONTINUE
*      WRITE(6,150)X ----->実行文となる
          :
          ↓
      OPTION ASTER
      100 CONTINUE
*      WRITE(6,100)A
          :
```