FORTRAN77 EXとベクトルプロセッサについて

総合情報処理センター
花田 英輔

1 はじめに

センターでは、新システム(ミニスーパーコンピュータVP1200システム)の導入に伴い、FORTRANコンパイラを、FORTRAN77の拡張版であるFORTRAN77 EXに変更した。

本稿では、従来のFORTRAN77に比べた拡張機能を中心に、新しいコンパイラを紹介し、次にベクトルプロセッサとFORTRANの関係に関して述べる。

2 FORTRAN77について

FORTRANは計算機用高級言語として1956年頃IBM社が開発して以来、現在に至るまで広く数値計算系に用いられており、この間何度かの仕様改定や標準化が行われている。

現在の標準は1978年にANSI(American National Standards Institute)が出した規格で、これをそのままISO(International Standard Organization)が認めて現在のFORTRAN77となっている。その後ISOが1980年に規格の見直しを始めて以来の改定作業が遅れており、1990年にようやく終了の目度がついた状態のようである。(新しい規格の通称はFortran 90とのこと。)

FORTRAN77に対する利用者からの要求も多くなり、メーカー側としてもそれに対処する必要があるため、同じFORTRAN77でもメーカー各社で拡張機能を追加しており、この企画外の部分に違いがある。

ただし、そもそも規格とは大枠を決めているだけなので、「拡張部分」については問題にならない。

センターでは、旧システム(FACOM M-760/30)導入時のFORTRAN77をそのまま利用してきた。(もちろん標準のFORTRAN77に対して富士通株式会社独自の拡張機能が付加されている。)

今回のFORTRAN77 EXへの拡張は、

- ベクトルプロセッサの導入などの環境の変化
- 新しい技術の導入による新しい機能の利用可能性拡大

といった理由に基づいて検討の結果、従来のFORTRAN77を入れ替え、導入したものである。

また、大型計算機センターで富士通製の汎用計算機を使用している大学(九州大学、京都大学、名古屋大学)では既にFORTRAN77 EXに移行している。

3 FORTRAN77 EXによる機能の拡張

一般にコンピュータ関係で機能拡張を行う場合、一つ前の機能はそのまま受け継ぐこと(これを「互換性を保つ」という)を目標とする。特にソフトウェアの場合は旧版利用者への負担を軽くする意味でも互換性は重要視されるが、互換性を完全に保つことができない場合もある。(互換性を保てなかった部分を「非互換部分」と呼ぶ。)

ここでは、FORTRAN77のEXへの拡張によって新たに追加される機能を中心に紹介するが、コンパイル、リンク、実行の方法については従来と全く同じである。

まず、FORTRAN77 EXの主な特徴は、

1. Fortran 90の機能の一部先行採用
2. 最適化技術の更新
3. 巨大プログラムへの対応
4. 新規開発ハードウェアへの対応

といったことが上げられる。

以下、特に直接プログラム作成と実行に関わることを中心に述べるが、詳細については、マニュアル類(参考文献に示す。いずれもセンターに常備)を参照されたい。

3.1 プログラム文法関係

1. 文の途中から注釈が書ける

   これまでは、注釈(コメント)は、独立した行に書かなくてはならなかったが、拡張により普通の行の途中に文字"!"を書くことで、!以後を注釈と見なすようになった。

(例)

```
      DO 10 I=1,10 !INITIALIZATION LOOP
        ARRAY(I) = 0
10    CONTINUE
```

2. 暗黙の型宣言を抑止可能とすることができる

これまでは、暗黙の型宣言により、

- I,J,K,L,M,N で始まる変数名は 4 バイト整数型
- その他の文字ではじまる変数名は実数型

となっていたが、これを抑止することができる。

(例)

```
IMPLICIT NONE !全ての暗黙の型宣言を抑止
```

3. 英文字とアンダーバー (アンダースコア) が使用できる

従来の FORTRAN プログラムでは、変数を始めほとんどを大文字で記述していたが、今回の拡張により小文字が使用できるようになりった。

また、アルファベット以外に"_"(アンダーバーまたはアンダースコアと呼ぶ) を変数名の中に使うことも可能である。

(例)

```
REAL a,B,a_1 !この場合 a ≠ a_1
```

4. 変数や定数の長さは最大 31 文字

これまでは最長 6 文字であったが、31 文字まで使える。(ただし余り長いとバグの素なので注意が必要。)

(例)

```
REAL CENTER_NO_DAREKA_NO_GENKOU
  !これは 26 文字なので OK
```

ただし、外部名 (サブルーチン名など) は先頭の 7 文字が有効。

5. 一文を継続する場合の最大値が増える

これまでは FORTRAN の一文として最大 19 行までであったが、99 行まで継続できるようになった。(しかし、余り長い文はバグの素なので勧めることはできない。) なお、1 文が 6600 文字を超えることはできない。

6. 関数、サブルーチンの引数の最大個数が増える

これまでは引数の最大個数は255個であったが、1000個まで可能になった。(但し余り多いとバグの素なので注意が必要。)

7. 文字定数の最大長が延びる

これまでの最大255文字までが最大32767文字までにまった。

8. 整数として採ることのできる値が大きくなる

これまでは、DOUBLE INTEGERを宣言しても4バイトまでだったが8バイトの整数が使えるようになった。(絶対値の最大は $2^{63}-1=7.9\times10^{18}=$ 約8百京)

9. 日本語が変数名、定数名として使えるようになる

これまで、プログラム中で日本語はほとんど使うことができなかったが、以下の様に使用可能になる。

a. 定数名、変数名として使える(ただし長さは15文字以内、日本語定数の長さは最大16383文字)なお、暗黙の型宣言では実数型に扱われる。

(例)

```
NCHARACTER*16383 定数        !長さ 16383 文字の「 定数」という名の定数
PARAMETER(定数=NC'総合情報処理センター')
NCHARACTER*5       変数
```

b.IMPLICIT文にも使える

(例)

```
IMPLICIT INTEGER(回)       !回で始まる名の変数は整数型
回数=10 !回数は回で始まるから整数型
```

c.WRITE文、FORMAT文中にも日本語編集記述子の指定ができる(N型)

(例)

```
100     FORMAT(N5,M1)
        WRITE(6,'(1H,N5)')   変数
```

d. シフトコード付きの入力が可能

(例)

```
READ(5,100) 変数
```

3.2 その他

今回の拡張にともない、C 言語との連絡がよくなり、引数を用いた call がどちらからも可能になった。

文法的にはこれまで以上に忠実なチェックが入りるので、プログラミング時には注意されたい。

さらに、メッセージ番号やエラー時に返す値が一部変わったのでマニュアル類を参照されたい。

4 ベクトルプロセッサについて

4.1 ベクトルプロセッサ

新システムのベクトルプロセッサは、当分の間 MSP でのみ利用可能である。

ベクトルプロセッサとは、繰り返し演算を高速に行うものと考えればよい。(以下、ベクトルプロセッサのことを VP と略す。)

VP を用いれば何でも実行速度が上がるというものではない。(新システムでは CPU 自体も変わったので全体的にも高速化された。) ファイルの入出力速度と VP は関係ないので、その点をよく理解して利用されたい。

4.2 VP を用いるための FORTRAN77 プログラミング

FORTRAN77 EX で書いた全てのプログラムが、VP の利用によって高速化される保証はない。次のような場合に効果がある。

- DO ループがある。
- 配列を使っており、かつ各要素に対して DO ループ内で計算をしている。

例えばプログラム中の次のような部分は VP を使うとかなり高速化される。

(例 1)

```
      DO 10 I=1,n
        A(I)=DBLE(I)*K
10    CONTINUE
```

(例 2)

```
      DO 30 J=1,10000
        DO 20 I=1,10000
          MULT(I,J)=DBLE(I)*A(I)
20      CONTINUE
30    CONTINUE
```

ベクトル化 (VP による実行時の高速化) される対象は上記例の通りであるが、DO ループ中に次のような文があった場合はベクトル化されない。

- 文番号代入分
- 計算型の GOTO 文
- 割当型の GOTO 文
- PAUSE 文
- ファイル終了指定子または誤り終了指定子を持つ入出力文
- 選択戻り指定子をもつ CALL 文
- RETURN 文
- STOP 文 (ただし DO ループ中に 2 回以上ある場合)

なお、DO ループ中に IF 文がある場合や、組み込み関数がある場合はほとんどの場合ベクトル化の対象になる。詳しくは参考文献を参照されたい。

4.3 VP を用いた高速化の効果

VP を用いることにより、どの程度高速化できるかは、次の式により説明される。

$$P = \frac{1}{(1-V)+\frac{V}{\alpha}}$$

ここで P はベクトル化しない場合の CPU 時間をベクトル化した場合の CPU 時間で割った値、即ち相対的な高速化比率である。

また、V はベクトル化率 (前命令中のベクトル化される命令数の比率)、αはベクトル／スカラー速度比である。αの値は、プログラム中のループの回数に関係する。

上記の通り、プログラム中でベクトル化される命令数を増やし、かつ DO ループの回数が多いほど高速化されることがわかる。

4.4 VPを使うには

VPを使う場合は、ジョブをジョブクラスGのバッチ処理とし、かつFORTコマンドにVP=YESというオペランドをつけることにより、VPを用いたコンパイルを指示する。

(例)

```
//F0000* CLASS=G
//EXEC FORT,VP=YES
...
```

なお、VP=YESをつけない場合にはVPは用いられず、FORTRAN77 EXのコンパイラによるコンパイルが行われる。

コンパイル結果を充分確認すると、ベクトル化された部分には、プログラム文の左にVが、されなかった部分にはSという文字が出力リスト上に表示される。Vの部分が多いほど高速化される。なるべく多くのDOループ部分にVが表示されるように工夫したプログラミングをお願いしたい。(この工夫をチューニングと呼ぶ)

チューニングのためにアナライザを用意している。ただし、このアナライザは実行結果を詳細に記述するのが第一目的であり、デバッグが目的ではない。利用方法は参考資料を参照されたい。

5 おわりに

以上の点を参考にし高速演算を必要とする方々は、VPを用いたジョブとすることを試みていただきたい。

新しいコンパイラ、VPにつきまして、ご利用になったご感想、ご意見をセンターにお寄せ下さい。

参考資料

○FORTRAN77 EXに関するもの

- 竹生 政資　新コンパイラ「FORTRAN 77 EX」について
  九州大学大型計算機センター広報　Vol.24 No.5 pp.523-540 1991.9
- 富士通株式会社　FORTRAN77 文法説明書 1991年6月版
  99SP-8032-1

- 富士通株式会社 OS IV/MSP FORTRAN77 EX 使用説明書
  V12用　79SP-5031-1

- 富士通株式会社 OS IV FORTRAN77 EX メッセージ説明書
  V12用　70SP-5321-1

○ベクトルプロセッサに関するもの

- 富士通株式会社 OS IV/MSP FORTRAN77 EX/VP 使用手引書
  V12用　79SP-5041-1

- 富士通株式会社 FORTRAN77 VP プログラミング手引書
  99SP-0080-1

- 富士通株式会社 OS IV/MSP アナライザ使用手引書　(FORTRAN77,VP用)
  V12L20用　79SP-5080-