

Mortran 3 (1)

平山 英夫

KEK, 高エネルギー加速器研究機構

はじめに

- Mortran – More Fortran
- Fortranのプリコンパイラー
- Mortranは、ANSI の FORTRAN77で書かれたプログラムである – mortran3.f or mortran3.for
- EGS4 は、Mortran3の機能を効率的に使用している
 - システムのコンパクト化
 - マクロを使用した改良

ステートメントの終わりはセミコロン (;)

- ステートメントは、任意の位置から書くことができる
- ステートメントは、特別な事をせずに、次の行に継続する事ができる
- 1行に複数のステートメントを書くことができる

Mortran3

X=1.0+XX; X=X+X*X;

YY=SQRT(X+5.6*X)+

X2.9-ALOG(X);**

Fortran

X=1.0+XX

X=X+X*X

YY=SQRT(X+5.6*X)+X2.9-ALOG(X)**

コメント:(") で囲まれた領域

- コメントは、変数内以外であれば、プログラム中どこにでも書くことができる
- 数行にわたるコメントを書くこともできる (%Q0 mode の時、(標準)).

Mortran3

"Example of COMMENT"

X=1.0+XX "XX is input data"; X=X+X*X;

YY=SQRT(X+5.6*X)+ "Continue to next line"

X**2.9-Alog(X);

FORTRAN

C Example of COMMENT

C XX is input data

X=1.0+XX

X=X+X*X

C Continue to next line

YY=SQRT(X+5.6*X)+X**2.9-ALOG(X)

DO-loop

- DO-loop は、DO I=J,K,N [.....;]と表現すれば良い
 - 文番号は不要
 - CONTINUE文は不要
 - J, K, N の機能は、FORTRANの DO-loop と同じ
- 次の例のような DO-loop を多重に使用する事ができる
 - “[” の数と“]”の数は一致しなければならない。

Mortran3

DO I=1,10 [DO J=1,20 [DO K=1.5 [A(I,J,K)=I*J*K;]]]

FORTRAN

DO 11 I=1,10

DO 21 J=1,20

DO 31 K=1.5

A(I,J,K)=I*J*K

31 CONTINUE

32 CONTINUE

21 CONTINUE

22 CONTINUE

11 CONTINUE

12 CONTINUE

IF Statement

- 基本となるIF statement は、IF e [.....;] である。
 - “e” は任意の論理表現で、ブロック内には、複数のstatementを書いても良い。
 - 論理表現は、()で囲んでも良いし、囲まなくても良い

IF A.LT.B [C=D; E=F;] G=H; or IF(A.LT.B) [C=D; E=F;] G=H;

- IF-ELSE statementは、次のように表現される。

IF e [.....;] ELSE [.....;]

- “THEN” や “ENDIF” は各必要がない
- IF-ELSEIF statementは、次のように表現される

IF p [...;] ELSEIF q [.....:] ELSEIF r [.....;] ELSE [.....:]

- “[” の数と “]” の数は、一致しなければならない

Mortran3

IF A1.LT.B1 [C=D1; E=F1;]

IF(A1.LT.B1) [C=D1;]

ELSEIF(A1.EQ.0.0) [E=F1;]

ELSE [C=0.0;]

FORTRAN

IF (A1.LT.B1) THEN

C=D1

E=F1

END IF

IF ((A1.LT.B1)) THEN

C=D1

ELSE IF ((A1.EQ.0.0)) THEN

E=F1

ELSE

C=0.0

END IF

入出力文

- 標準的なFORTRANの入出力文[READ(5, とWRITE(6,)]は、次のように表現できる:

INPUT I-o list; (format list); READ(5, に対応

例: INPUT NCASES; (I6);

OUTPUT I-o list; (format list); WRITE(6, に対応

例: OUTPUT IXX; (' IXXST=',I12);

- 書式部の文番号は不要
- 書式部は、個々のINPUT あるいは OUTPUTに常に必要.

Mortran3

INPUT XX;(F10.3)

FORTRAN

READ(5,10) XX

10 FORMAT(F10.3)

Mortran3

OUTPUT X,Y;

(' X=',G15.5,' and YY=',G15.5);

FORTRAN

WRITE(6,20)

20 FORMAT(' X=',G15.5,' and YY=',G15.5)

その他の入出力UNIT

- その他の入出力UNITを使用する場合

READ(1,:FORMAT1:) I-o list; :FORMAT1:FORMAT(format list);

WRITE(7,:FORMAT7:) I-o list; :FORMAT7:FORMAT(format list);

- :FORMAT1:, :FORMAT7: は、後で紹介するlabel
- FORTRANの場合と同様に、この形の入出力文では、書式部は、1回定義するだけでよい。
- READ(5, や WRITE(6, の場合にこの方法を使用しても良い

Multiple Assignment

- 複数の変数に同じ値を割り当てたい場合には、次の表現を使う事ができる

/p,q,r,.....,z/=e;

- 変数の型は同じである必要はない

Mortran3

/I, A(I,K), J/=SQRT(X/2.0);

FORTRAN

I=SQRT(X/2.0)

A(I,K)=SQRT(X/2.0)

J=SQRT(X/2.0)

- Multiple assignmentは、変数を初期化する場合に便利である

Label

- “:” で囲まれた部分はlabelと認識される

例 **:USER-PHOTON-DISCARD:.....;**

- Labelで使用する文字数に制限はない
- 従って、labelの意味を理解しやすい名前を使用する方がよい
- label は、FORTRANでは、文番号に置き換えられる

Operator

- relational operatorsとしては、FORTRANで使用される .LT., .EQ., 等の他に以下の記号を使用できる

<, <=, =<, =, ~=, =>, >=, >

- logical operatorsとしても .AND., .OR.等の他に以下の記号を使用する事ができる

&, |, ~

- 一つのstatement中で、FORTRANで使われる表現と記号を混同してはいけない

IF(A>B & C>D) or IF(A.GT.B.AND.C.GT.D) OK

IF(A>B.AND.C>D) is not OK!

Mortran, FORTRAN リスト出力のコントロールコマンド

- %で始まるコマンドは1カラムから書かなければならない
- !で始まるコマンドは、どの場所にも書いてもよい
- %E: Mortran リストでページ替えをする
- %L or !LIST; : Mortran リストを出力する
- %N or !NOLIST; : Mortran リストを出力しない
- %In or !INDENT Mn; : Mortran リストでネスト毎にn-文字分インデントを行う
- %F: FORTRANモードに切り替える。%Fと%Mを使いFORTRAN statement とMortran statementを併せて使用することができる。

%Fのバグ: Mortran3には、%F の直前のstatementが正しく変換されないというバグがある。これをさけるために、%Fの前に、;を含むダミーstatementを入れる必要がある

- %M: Mortranモードに切り替える
- !INDENT Fn;:FORTRANリストでネスト毎にn-文字分インデントを行う

Mortran, FORTRAN リスト出力のコントロールコマンド

- !COMMENTS;: MortranのコメントをFORTRANリストにコメントして出力する
- !UNCOMMENTS;: MortranのコメントをFORTRANリストに出力しない
- %An: MortranのコメントをFORTRANリストに出力する際のコントロールコマンド
 - n=0: FORTRANリストに出力しない。!NOANNOTATE;と同じ
 - n=1: 2 から80カラムにコメントとして出力する。!ANNOTATE;と同じ
 - n=2:40 から80カラムにコメントして出力する
- %Qn: Mortranのコメントのコントロールコマンド(default is 0)
 - n=0: コメントの最初と最後には”が必要。複数行にわたるコメントが可能。
 - n=1: 各行の最後の”は不要。コメント行は、行の最後で終了したと見なされる。複数行にわたるコメントは使用不可。

Mortran使用時のエラー

- Statementの最後に“;”を書くことを忘れた場合
 - Statementの最後に“;”を忘れると、2つのstatementが一つのstatementと見なされる。Mortranではエラーにならないが、FORTRANコンパイル時にエラーが生じる。

Mortran

X=X+1 Y=Y+1

Y=Y+1;

FORTRAN

X=X+1 Y=Y+1

***MISPLACEDSYMBOL(=) * ?**

Mortran使用時のエラー

- [,] や ” の過不足
 - [と] の数は、等しくなければならない。等しくない場合には、Mortranエラーが生じる。(mortjob.mlt[PC], mortjob.list[Unix]を調べる)

```
0 DO I=1, 100 [  
1 DO J=1,3 [  
2 Z=Z+I*J;  
2 ]  
1 STOP;  
1 END;
```

Nest level

END OF MORTRAN INPUT
UNCLOSED BLOCKS

- エラーの場所を見つけるには、
 - プログラム単位の最後に %E を書いておく
 - mortjob.mlt(又は、mortjob.list)で、%Eを順に検索し、その位置でのネストレベルを調べる
 - もし、ネストレベルが0でない場合には、]が余分 (-の時) または不足 (+の時) がそのプログラム単位で起きている

Mortran使用時のエラー

-)の過不足

Mortran

```
0 DO I=1, 100 [  
1 DO J=1,3 [  
2 IF(I.LT.10 [  
3 Z=Z+I*J; ]  
2 ]]  
0 STOP;
```

FORTRAN

```
DO 11 I=1,100  
DO 21 J=1,3  
IF(I.LT.10  
* INSUFFICIENT PARENTHESSES * ?  
* INVALID STATEMENT * ?  
Z=Z+I*J  
21 CONTINUE  
22 CONTINUE  
11 CONTINUE
```

Mortran使用時のエラー

- %Q1 モードの時以外は、コメントは必ず最後に ”を書かなければならない (mortjob.mlt[PC], mortjob.list[Unix])

Mortran

0 "Comment must be closed

" 0

" 0 IF A1.LT.B1 [C=D1; E=F1;]

" 0

" 0 IF(A1.LT.B2) [C=D2;]

" 0 ELSEIF(A1.EQ.0.0) [E=F2;]

" 0 ELSE [C=0.0;]

" 0 STOP;

" 0 END;

END OF MORTRAN INPUT

UNCLOSED COMMENT

83 PERCENT RULE CAPACITY REMASINIG

1 MORTRAN ERRO(S)

Indicate that
comment is
continuing.

