

新コンパイラ「FORTRAN 77 EX」について

竹生, 政資
九州大学大型計算機センター研究開発部

<https://doi.org/10.15017/1470201>

出版情報 : 九州大学大型計算機センター広報. 24 (5), pp. 523-540, 1991-09-25. 九州大学大型計算機センター
バージョン :
権利関係 :



新コンパイラ「FORTRAN 77 EX」について

竹生 政資 *

1. はじめに

本稿では、九州大学大型計算機センターにおいて近々公開予定である新しい FORTRAN コンパイラの特徴および従来のコンパイラとの互換性について解説する。新しいコンパイラは、スカラー計算のためのオブジェクトコードを生成する「FORTRAN 77 EX」とベクトル計算のためのコードを生成する「FORTRAN 77 EX/VP」の2つからなり、それぞれ従来のコンパイラ「FORTRAN 77」および「FORTRAN 77/VP」を拡張したものである(表1)。以下では簡単のため、特にスカラー計算用とベクトル計算用を区別する必要がないかぎり、2つの新コンパイラをまとめて単に「FORTRAN 77 EX」と記することにする。

FORTRAN(FORMular TRANslator)は1950年代半ばにIBM社のバックス(Backus, J.W.)らによって当時の大型計算機 IBM704 のために開発された科学技術数値計算言語である。その後ほぼ10年ごとに言語仕様が改訂され、近代的プログラミング言語として着実に進化してきた。言語仕様の改訂に際しては、既存の仕様(少なくともひとつ前の仕様)と互換性を保つよう細心の注意を払いつつ情報科学の進歩を積極的に吸収し、今日かなりの数のプログラミング言語の中にあつて最も長い実用の歴史を誇っている。FORTRAN 言語の体系は

- (a) 言語仕様規格
- (b) 処理系(コンパイラ)
- (c) ライブラリ

の3つからなっている。

言語仕様規格は文法の大枠について規定したものである。文法はできるだけコンピュータのハードウェアから独立した仕様になっている。例えば、メモリの番地という概念は変数に置き換えられ、メモリに格納されるデータの種類として整数型や実数型のようなより抽象度の高い概念が用いられている。FORTRAN 言語仕様の最初の国際推薦規格は、1966年 ASA によって規格化された Basic FORTRAN と Full FORTRAN (当時業界で使われていた FORTRAN II と FORTRAN IV に基づく) に、さらに ECMA から提案された Intermediate FORTRAN を加えた三つとして、1972年 ISO により制定された。これは通称 FORTRAN 66 と呼ばれている。これを基にして日本では JIS FORTRAN (水準3000, 水準7000, 水準5000) が制定された(水準の高いものは低いものを含む)。ひき続き次の言語仕様改訂のための作業が開始され、文字型データや構造化プログラミングのための機能、直接アクセスのファイル入出力機能などを追加した FORTRAN 77 の規格が 1978年 ANSI によって制定された[1]。これはそのままの形で国際規格として承認され 1980年 ISO 規格となった。FORTRAN 77は full language と subset language の2つの水準からなっている。これに対応して日本では 1982年 JIS X3001-1982 FORTRAN (基本水準, 上位水準) が制定された。現在使用されている FORTRAN 処理系のほとんどが FORTRAN 77 の言語仕様に基づいている。FORTRAN 77 の制定後、ひき続きアメリカ(ANSI)を中心に ISO 各国が協力して次の新しい規格改訂のための作業が続けられてきた。新しい規格は当初 Fortran 8X と呼ばれ1980年代の規格化を目標としていたが、1990年によく改訂作業がほぼ完了し、通称を Fortran 90 (FORTRAN 90ではなく)とすること

平成3年7月25日受理

* 九州大学大型計算機センター 研究開発部

になった[2]。文献[3]によれば、「プログラム言語の原案の開発に関して、これほど多数のメンバーが、これほど長期にわたり、これほど積極的に協力して活動している例は他にないであろう」。Fortran 90の概要については第5節で簡単に紹介する。

- (注) ASA: American Standards Association (米国規格協会, ANSIの前身)
 ISO: International Standards Organization (国際標準化機構)
 JIS: Japanese Industrial Standard (日本工業規格)
 ANSI: American National Standards Institute (米国規格協会)
 ECMA: European Computer Manufacturing Association (欧州電子計算機工業会)

処理系は文法にしたがって記述されたソースプログラムをハードウェアが実行できる形に変換するためのものである。Fortran 処理系は、教育用やデバッグ専用のものを除き伝統的にコンパイラとして作られてきた。一般に、処理系には処理形態の違いによりコンパイラとインタプリタがある。処理系はハードウェアの性能を最大限に引き出すよう最適化技術を駆使して設計されるため、完全にハードウェアに依存する。世の中には、パソコン、ワークステーション、ミニコン、汎用大型計算機、スーパーコンピュータとさまざまな種類のコンピュータがあるが、それぞれに対応して種々の Fortran コンパイラが存在する。表1に当センターにおける FORTRAN コンパイラ(富士通製)の変遷を示した。

表1. FORTRAN コンパイラの変遷

汎用大型計算機	ベクトル計算機
・ FORTRAN C/D (1969/4 ~1977/9) ・ FORTRAN H (1975/4 ~1977/9) ・ FORTRAN IV GE/HE (1977/11~1984/5) ・ FORTRAN 77 (1980/5 ~1992/) ・ FORTRAN 77 EX (1991/ 未定)	・ FORTRAN 77/VP (1985/12~1992/) ・ FORTRAN 77 EX/VP (1991/ 未定)

ところで、言語仕様規格はあくまでも文法の大枠を定めたものであり、実際に処理系メーカーから提供される言語仕様は、例えば同じ FORTRAN 77 の規格に基づいた処理系であっても、細部においていろいろと違いがある。全国共同利用大型計算機センターで運用されている FORTRAN 77 処理系の差異については詳しい比較表が作成されている[4]。

Fortran ライブラリはそれぞれの用途のサブルーチンや関数副プログラムの集合(コンパイルされたオブジェクトモジュールの集合)である。コンパイラに標準装備されている組み込み関数をはじめ、富士通(株)製の数値計算ライブラリ SSL, SSL II, 名古屋大学で開発された数値計算ライブラリ NUMPAC, 利用者個人の私用ライブラリなどがある。このほかにも図形処理や統計計算など、FORTRAN には膨大なソフトウェア資産がある。

Fortran の言語仕様や処理系は情報科学やハードウェアの進歩を反映して時代とともに徐々に改革されてきたが、改訂の際に問題になるのが過去のソフトウェア資産との互換性である。言語仕様や処理系の設計に携わる者は、少なくともひとつ前の仕様とは完全な互換性を保つことを目標に改訂を行なうが、現実にはどうしても吸収しきれない部分(非互換部分)が残る。この結果、

改訂前の処理系では問題なく動作していたプログラムが、新しい処理系では正常に動作しなくなる場合が起こりうる。これについては利用者が個別に対応せざるをえない。今回のFORTRAN 77 から FORTRAN 77 EX への改訂に際しても非互換部分がいくつか存在する。この問題については第4節で詳しく述べる。非互換の問題は、単にソースプログラムレベルだけでなくオブジェクトモジュール（過去に翻訳作成したライブラリ）レベルでも発生するので注意が必要である。

2. FORTRAN 77 EXの特徴

新しいコンパイラ FORTRAN 77 EX および FORTRAN 77 EX/VP は、従来の FORTRAN 77 および FORTRAN 77/VP に較べて大幅に機能が拡張されている。主な特徴として、

- (1) 新しい言語規格 Fortran 90 の先行採用（一部分のみ）
- (2) 最新の最適化技術、ベクトル化技術の採用
- (3) IBM社の VS FORTRAN との互換性強化
- (4) 巨大プログラムへの対応
- (5) 新しく開発されたハードウェア（システム記憶装置、ディスクアレイ）への対応

などがあげられる。

特にシステム記憶装置(SSU: System Storage Unit)は、主記憶と補助記憶装置（磁気ディスク装置）の間に位置づけられる新しいハードウェアで、直接プログラムから操作することはできないが半導体(D-RAM)でできており、高速の補助記憶装置として今後大規模配列数値計算にとって不可欠なものとなりつつある[5]。

また、上でも述べたように、Fortran 90 はまだ規格が完成したばかりであり、今回のコンパイラ改訂では残念ながらその一部仕様だけしか実現されていないが、FORTRAN 77 EX はここ数年内に完全な Fortran 90 仕様へと段階的に発展していぐものと思われる。

3. FORTRAN 77 EXの機能強化項目

新コンパイラの主な機能強化項目について簡単に紹介する。詳細については FORTRAN 77 EX 文法書および FORTRAN 77 EX 使用手引書[6, 7, 8]を参照されたい。

3.1 言語仕様の拡張

(1) Fortran 90 の先行採用（一部分）

- ・文字 "!" によって行の途中から注釈が書ける。
- ・暗黙の型宣言を抑止できる。
- ・英小文字およびアンダースコア（下線）が使用できる。
- ・名前（変数名や定数名など）の最大長は31文字。

(例)

program main	! 行の後ろのコメント
implicit none	! 暗黙の型宣言抑止
real a, b, c_ext	! 英小文字, アンダースコアを使った変数名
real x12345678	! 名前の最大長は31文字
end	(ただし外部名の場合は先頭7文字が有効)

(2) コンパイラ定量制限の緩和

- ・ 1 文の継続可能行数が従来の 19 行から 99 行に拡大。
- ・ 関数やサブルーチンの引数の最大個数が 255 個から 1000 個へ拡大。
- ・ 文字定数の最大長が 255 文字から 32767 文字へ拡大。

(3) 8 バイト整数型の導入

- ・ 絶対値が $(2^{63} - 1)$ 以下の整数が処理可能。

3.2 アドレス拡張域の利用による大規模プログラムへの対応

旧コンパイラでは、翻訳あるいはGOオプションによる翻訳・実行は基本領域（OSの作業域も含めて最大16MB）だけに限られていたが、新コンパイラではこれらがアドレス拡張域（最大2GB）で実行できるようになり、従来は処理できなかった大規模プログラムの処理が可能となった。

3.3 C言語との言語間結合機能

FORTRAN とC言語の間で引数渡しによりお互いの機能を参照することができる。

(例) Cの関数を FORTRAN から参照する。

```
PROGRAM FORT
J=1972
K=1990
.....
CALL JLIINT      !環境切り替え文
CALL FUN(J,K)
CALL JLIEND      !環境切り替え文
.....
END
```

```
fun(ii,jj)
int *ii,*jj;
.....
printf("* FUN *");
}
```

(例) Cから FORTRAN の関数を参照する。

```
main()
{
    int fun();
    .....
    jlmsrt(0x80000000); /*環境切り替え文*/
    k=func1(&l);
    jlimend(0);         /*環境切り替え文*/
    .....
}
```

```
FUNCTION FUNC1(L)
.....
RETURN
END
```

3.4 日本語機能の強化

- ・ 15 文字以下の日本語名標（定数名や変数名など）が使用できる。
- ・ IMPLICIT 文による暗黙の型宣言に日本語が使用できる。

- ・文字書式による日本語編集記述子の指定が可能。
- ・シフトコード付きデータ入力が可能。
- ・日本語文字定数の最大長は1 6 3 8 3 文字。
- ・日本語によるコンパイラ出力情報。

(例)

IMPLICIT INTEGER(回)	! 日本語文字による暗黙の型宣言
NCAHRACTER*16383 日本語定数	! 文字定数長として16383文字確保
PARAMETER(日本語定数=NC' あいうえお')	
NCAHRACTER*5 日本語変数	
回数=10	! IMPLICIT文により回数は整数型
READ(5,100) 日本語変数	! シフトコード付きデータ入力
100 FORMAT(N5,M1)	
WRITE(6,'(1H ,N5,M2)') 日本語変数	! 日本語編集記述子
END	

3. 5 索引入出力機能の強化

- ・交替インデクスのサポートにより、1つのファイルに異なる複数のキーでアクセス可能。
- ・キー検索の方法が追加され、指定したキーと一致する記録だけでなく、次に大きいキーにもアクセス可能となった。
- ・キーなしや重複キーの誤り状態を検出し分岐先を指定できる。

3. 6 高速入出力機能の強化

高速入出力機能は、主記憶に収まりきらない巨大なデータをバッファを使用することなく複数の直接アクセス記憶装置へ並列に入出力する機能である。

- ・入出力並びは従来配列だけだったが新たに変数も可能になった。
- ・最大ブロック長が従来の3 2 K Bから4 7 K Bに拡大。
- ・FORTRAN 記録の最大長が従来の1 M Bから無制限になった（ただし、MSP AF II 以降）。
- ・従来は実行時オプション"HIO"で指示していたのが、DD文で指示するようになった。
- ・従来の順次入出力文のほかに、直接入出力文および非同期入出力文も処理の対象となった。

3. 7 VIO/F機能のSSU対応

SSU(System Storage Unit)は主記憶と磁気ディスク装置との間に位置づけられる新しいシステム記憶装置である。ジョブ制御文のDD文(SUBSYSパラメタ)でVIO/Fファイル(Virtual I/O File)を定義することにより、SSU上に展開されたVIO/Fファイルを一時データセットとみなし、入出力処理のためのバッファを用いず、主記憶とSSUとの直接データ転送により高速に入出力処理を行なう。これにより磁気ディスク装置によるI/Oに比べて経過時間がはるかに短縮される。

3. 8 ライブラリの全静的結合機能

ロードモジュール作成時にリンケージエディタにより結合されるFORTRANライブラリにはレジデントライブラリとランジェントライブラリの2種類がある。従来の静的結合では前者のライブラリだけが静的に結合され後者は実行時に動的に結合されていた。新FORTRANシステムでは結

合編集時に新しいライブラリを選択することにより全ライブラリを静的に結合したロードモジュールを作成することができる。これにより2種類のライブラリの世代間の相違を考慮する必要がなくなり、アプリケーションプログラムの流通性が向上する。詳細については[7]の“2.1.5 結合編集と実行”を参照されたい。ただし、SSL IIを使用したプログラムは言語間結合用ロードモジュールを動的に結合するため、実行時にトランジェントライブラリが必要になる。

3. 9 その他

- ・プログラム診断機能。

演算桁数を変化させることにより演算結果の精度を診断

許容値を認めない関係演算の検出

実引数のオーバーラップや整数型データ演算におけるオーバーフローの検査

- ・エディタとの連携機能。

- ・実運用プログラムでの会話型デバッグ機能。

- ・IBM社のVS FORTRANとの互換性強化。

- ・副プログラムのインライン展開の最適化を強化。

- ・ループアンローリングの展開を従来の2重化のみから最高9重まで拡張。

4. FORTRAN 77とFORTRAN 77 EXの非互換項目

新コンパイラは旧コンパイラに比べて文法チェックが非常に厳しくなっている。このため、従来のFORTRAN 77では一見問題なく動作していたプログラムでもFORTRAN 77 EXではエラーメッセージが出て実行が中断される可能性がある。また、何のエラーメッセージも出ないが結果が異なる可能性もある。移行に際しては十分チェックする必要がある。

さらに、新コンパイラのエラー識別番号体系は旧コンパイラのものとは全面的に異なっているの[9]、プログラム中でエラー識別番号を直接引用している箇所は書き換える必要がある。

以下、非互換項目の主なものについて説明する。

(1) ネームリスト入力データ中に空項目が出現したとき動作が異なる。

- ・FORTRAN 77 の動作

空項目に対応する並びに0を代入する。

- ・FORTRAN 77 EX の動作

空項目に対応する並びの値は変わらない。

(例)

```

INTEGER*4 A(3) /9999,9999,9999/
NAMELIST/NN/A
READ(5,FMT=NN)
.....
入力データ
&NN
A=100,,300
&END
    
```

この例の場合、結果は次のようになる。

FORTRAN 77 では A(2)=0

FORTRAN 77 EX では A(2)=9999

- ・非互換の検出方法

一般的な検出方法はない。

- ・非互換の回避方法

入力データの空項目を 0 に修正するか、または入力並びを 0 で初期化しておく。

(2) OPEN 文による動的割り当てエラー発生時に IOSTAT に返却される値が異なる。

- ・FORTRAN 77 の動作

8 (動的割り当てルーチンの復帰値) を返却。

- ・FORTRAN 77 EX の動作

23 (メッセージ番号) を返却。

(例)

```
INTEGER IOS
OPEN(1, FILE='FORT.DATA', IOSTAT=IOS)
.....
```

この例で、動的割り当てエラーが発生したとき、結果は次のようになる。

FORTRAN 77 では IOS=8

FORTRAN 77 EX では IOS=23

- ・非互換の検出方法

JWE 0023I-S メッセージ (動的割り当てエラー発生) が出力される。

- ・非互換の回避方法

IOSTAT 指定子の値の判定値を変更する。

(3) OPEN 文の FILE 指定子で指定したデータセットの有無と、STATUS 指定子で指定した値に矛盾がある場合、IOSTAT に返却される値が異なる。

- ・FORTRAN 77 の動作

8 (動的割り当てルーチンの復帰値) を返却。

- ・FORTRAN 77 EX の動作

105 (メッセージ番号) を返却。

(例)

```
INTEGER IOS
OPEN(1, FILE='FORT.DATA', IOSTAT=IOS, STATUS='NEW')
.....
```

この例では、データセット"FORT.DATA" が既存の場合、結果は次のようになる。

FORTRAN 77 では IOS=8

FORTRAN 77 EX では IOS=105

- ・非互換の検出方法

JWE 0105I-E メッセージ (STATUS 指定子の値がデータセットの状態と矛盾) が出力。

・非互換の回避方法

IOSTAT 指定子の値の判定値を変更する。

(4) 索引 READ 文の実行時にファイル終了記録を検出した場合, IOSTAT に返却される値が異なる。

・FORTRAN 77 の動作

158 (メッセージ番号) を返却。

・FORTRAN 77 EX の動作

-1 (EOD 検出) を返却。

(例)

```

INTEGER*4 IOST, EOD
CHARACTER CH40*40
OPEN(1, ACCESS='KEYED')
EOD=158
READ(1, KEY='KEY1', FMT='(A40)') CH40
DO 100 I=1, 99
  READ(1, FMT='(A40)', IOSTAT=IOST) CH40
  IF (IOST.EQ.EOD) GO TO 10
100 CONTINUE
CLOSE(1)
.....
10 WRITE(6, *) 'EOD ERROR'
.....

```

この例の場合, 結果は次のようになる。

FORTRAN 77 では IOST=158

FORTRAN 77 EX では IOST=-1

・非互換の検出方法

一般的な検出方法はない。

・非互換の回避方法

判定する値を -1 に変更する。

(5) 直接処理の索引 READ 文, 直接処理の REWRITE 文および DELETE 文の実行時に, 指定したキーをもつレコードが存在しないとき IOSTAT に返却される値が異なる。

・FORTRAN 77 の動作

158 (メッセージ番号) を返却。

・FORTRAN 77 EX の動作

24 (メッセージ番号) を返却。

(例)

```

INTEGER*4 IOST, NOKEY
CHARACTER CH40*40

```

```

OPEN(1, ACCESS='KEYED')
NOKEY=158
READ(1, KEY='KEY1', FMT='(A40)', IOSTAT=IOST) CH40
IF(IOST.EQ.NOKEY) GO TO 10
REWRITE(1, FMT='(A40)', IOSTAT=IOST) CH40
IF(IOST.EQ.NOKEY) GO TO 10
DELETE(1, KEY='KEY2', IOSTAT=IOST) CH40
IF(IOST.EQ.NOKEY) GO TO 10
CLOSE(1)

.....
10 WRITE(6,*) 'NOT FOUND KEY ERROR'

```

この例の場合、結果は次のようになる。

FORTRAN 77 では IOST=158

FORTRAN 77 EX では IOST=24

- ・非互換の検出方法
一般的な検出方法はない。
- ・非互換の回避方法
判定する値を 24 に変更する。

(6) 索引 WRITE 文および REWRITE 文の実行時に、指定したキーをもつレコードが既に存在するとき IOSTAT に返却される値が異なる。

- ・FORTRAN 77 の動作
159 (メッセージ番号) を返却。
- ・FORTRAN 77 EX の動作
25 (メッセージ番号) を返却。

(例)

```

INTEGER*4 IOST, DUPKEY
OPEN(1, ACCESS='KEYED')
DUPKEY=159
WRITE(1, FMT='(A40)', IOSTAT=IOST) 'KEY1 WRITE'
IF(IOST.EQ.DUPKEY) GO TO 10
REWRITE(1, FMT='(A40)', IOSTAT=IOST) 'KEY2 REWRITE'
IF(IOST.EQ.DUPKEY) GO TO 10
CLOSE(1)

.....
10 WRITE(6,*) 'DUPLICATE KEY ERROR'

.....

```

この例の場合、結果は次のようになる。

FORTRAN 77 では IOST=159

FORTRAN 77 EX では IOST=25

- ・非互換の検出方法
一般的な検出方法はない。
- ・非互換の回避方法
判定する値を 25 に変更する。

(7) リスト指示 READ 文において、文字型以外の配列に対して文字データの入力を行なったとき、動作が異なる場合がある。

- ・FORTRAN 77 の動作
文字列の残りに空白を設定。
- ・FORTRAN 77 EX の動作
文字列の残りの値は変わらない。
JWE 0182I-W メッセージ（文字型以外の並びに対する定数が文字型）が出力。
(例)

```

INTEGER*4 I(3)
READ(5,*) I
.....
入力データ
'ABCDEFGF'
    
```

この例の場合、結果は次のようになる。

FORTRAN 77				
I(1)	A	B	C	D
I(2)	E	F	G	
I(3)				

FORTRAN 77 EX				
I(1)	A	B	C	D
I(2)	E	F	G	
I(3)	そ	の	ま	ま

- ・非互換の検出方法
JWE 0182I-W メッセージ（文字型以外の並びに対する定数が文字型）が出力された場合、この非互換の影響を受けている可能性がある。
- ・非互換の回避方法
入力データの長さを配列全体の長さと等しくし、残りに空白を付加するまたは、配列全体空白で初期化しておく。

(8) 論理IF文、DO WHILE 文、DO UNTIL 文、またはブロックIF文の論理式において
論理式の括弧の中と外に AND がある
論理式に NOT が現れ、その一次子が配列要素でかつその論理式に関係がある
のいずれかの条件を満たし、かつその中で論理値（真、偽）以外の値が使われているとき、
論理式の結果が異なる場合がある。

FORTRAN 77 EX では FORTRAN 77 より論理式の評価の最適化を強化しているためこのような非互換が生じる。

(例)

```
LOGICAL L1/ZF0000000/
LOGICAL L2/Z00000000/
LOGICAL L3/Z0000000F/
LOGICAL LA(2)/2*Z000F0000/
I=1
IF( (L1.AND.L2) .AND. L3 ) GO TO 2
IF( .NOT.LA(2) .AND. I.EQ.1 ) GO TO 3
.....
```

この例の場合、最適化の方法の違いにより結果は次のようになる。

・FORTRAN 77 の動作

以下のように、いずれも論理式を最終評価後に分岐する。

```
t1=L1.AND.L2
t2=t1.AND.L3
IF( t2.ne.0 ) GO TO 2
t3=.NOT.LA(2)
t4=I.EQ.1
t5=t3.AND.t4
IF( t5.ne.0 ) GO TO 3
```

・FORTRAN 77 EX の動作

以下のように、論理式を最終評しないで途中で分岐する。

```
IF( L1.eq.0 ) GO TO α
IF( L2.eq.0 ) GO TO α
IF( L3.ne.0 ) GO TO 2
α CONTINUE
IF( I.NE.1 ) GO TO β
IF( LA(2).eq. 0 ) GO TO 3
β CONTINUE
```

・非互換の検出方法

一般的な検出方法はない。

・非互換の回避方法

論理値でない値をもつ論理演算は、整数演算に修正する。

(9) 実行時のエラー識別番号体系が異なるため、エラー識別番号を引数として利用するエラー処理サブルーチンの実行結果が異なる場合がある。

(例)

```
INTEGER*4 ERRNO/200/, MPRINT/-1/
CALL ERRSET(ERRNO,0,MPRINT)

.....
CLOSE(10)
WRITE(10,*)
```

この例の場合、結果は次のようになる。

FORTRAN 77 では JZL200I-E メッセージ（補助入出力文の次の許されない入出力文を実行しようとした）の出力が抑止される。

FORTRAN 77 EX では JWE 0111I-E メッセージ（～文は～文の直後に実行できない）が出力される。

・非互換の検出方法

JWE 0305I-W メッセージ（エラー識別番号のサブルーチンが実行された）が出力される。

・非互換の回避方法

FORTRAN 77 系のエラー識別番号を FORTRAN 77 EX 系のものに修正する。

(10) 文字型の入出力並びに対し、L, I, F, E, D, Q 形編集を行なった場合、動作が異なる。

・FORTRAN 77 の動作

W レベルの診断メッセージを出力し、編集記述子に従った処理を行なう。

・FORTRAN 77 EX の動作

E レベルの診断メッセージを出力し、以降の入出力並びの処理を行なわない。

・非互換の検出方法

JWE 0151I-E メッセージが出力される。

・非互換の回避方法

文字型の入出力並びに対しては、A, G または Z 形編集記述子に修正する。

(11) 同一装置参照番号に対して、REWIND 文または ENDFILE 文（マルチファイル）なしで、書式付き入出力文と書式なし入出力文が混在した場合の動作が異なる。

・FORTRAN 77 の動作

正常終了する。

・FORTRAN 77 EX の動作

JWE 0113I-E メッセージを出力し、文を無視する。

・非互換の検出方法

JWE 0113I-E メッセージが出力される。

・非互換の回避方法

同一装置参照番号に対しては、書式付き入出力文と書式なし入出力文が混在しないよう、入出力文の書式指定を修正する。

(12) 区分編成ファイルの既存メンバに対し、READ 文の後に WRITE 文を実行した場合の動作が異なる。

- ・ FORTRAN 77 の動作
正常終了する。
- ・ FORTRAN 77 EX の動作
JWE 0112I-E メッセージを出力し、WRITE 文を無視する。
- ・ 非互換の検出方法
JWE 0112I-E メッセージが出力される。
- ・ 非互換の回避方法
WRITE 文を削除する。

(13) 文字書式中にエラーを検出したときの動作が異なる。

- ・ FORTRAN 77 の動作
書式仕様中に無効な文字を検出した場合および書式仕様が左括弧で開始していない場合はエラーとするが、その他の書式仕様の誤りを検出した場合は、FORTRAN の解釈を行い処理を続行する。
- ・ FORTRAN 77 EX の動作
書式仕様中に無効な文字を検出した場合および書式仕様が左括弧で開始していない場合はエラーとし、かつ次の場合もエラーメッセージを出力し、以下の処理を行なう。
 - ・ 書式の入れ子が 8 以上のとき、7 を越えた入れ子を無視して処理を続行。
 - ・ 編集記述子の指定に誤りがあるとき、書式仕様の最後の右括弧と見なす。
 - ・ 書式仕様の中の反復子に誤りがあるとき、書式仕様の最後の右括弧と見なす。
 - ・ 書式仕様が右括弧で終了していないとき、右括弧を補い処理を続行。
- ・ 非互換の検出方法
JWE 0154I-W メッセージ（書式の入れ子が 8 以上）。
JWE 0155I-E メッセージ（編集記述子の指定に誤りがある）。
JWE 0156I-E メッセージ（書式仕様の中の反復子に誤りがある）。
JWE 0158I-W メッセージ（書式仕様が右括弧で終了していない）。
以上のどれかが出力された場合、この非互換の影響を受けている可能性がある。
- ・ 非互換の回避方法
文字書式仕様中の誤った書式を正しい書式に修正する。

(14) OPEN 文の ACTION 指定子に WRITE を指定したとき BACKSPACE 文の動作が異なる。

- ・ FORTRAN 77 の動作
BACKSPACE 処理を行なう。
- ・ FORTRAN 77 EX の動作
JWE 0112I-E メッセージを出力し、文を無視する。
(例)

```
OPEN(1, ACTION='WRITE', FORM='UNFORMATTED')
WRITE(1) A
BACKSPACE 1
.....
```

この例の場合、結果は次のようになる。

FORTRAN 77 では BACKSPACE 処理を行なうかまたは入出力エラーとなる。

FORTRAN 77 EX では JWE 0112I-E メッセージを出力し、文を無視する。

- ・非互換の検出方法

JWE 0112I-E メッセージが出力される。

- ・非互換の回避方法

OPEN 文の ACTION 指定子を BOTH に変更する。

(15) 浮動小数点アンダフローエラーのエラーレベルが異なる。

- ・FORTRAN 77 の動作

JZL 208I-W メッセージが出力される。このエラーに対するエラー打ち切り回数は無制限のため、実行は打ち切られない。

- ・FORTRAN 77 EX の動作

JWE 0012I-E メッセージを出力し、エラーが10回発生した時点で実行を打ち切る。

- ・非互換の検出方法

JWE 0012I-E メッセージが出力される。

- ・非互換の回避方法

ERRSET サービスサブルーチンを使用してエラー打ち切り回数を無制限にし、メッセージ最大印刷回数を0にする。

(16) 母国語型（日本語型）の引数をもつ組み込み関数(LEN, INDEX)の結果が異なる。

- ・FORTRAN 77 の動作

母国語型（日本語型）の引数をもつ組み込み関数は文法では定義されていないが、翻訳時にエラーが出力されずに占有領域単位の値が返却される。日本語文字はそれぞれ2文字として数えられる。

- ・FORTRAN 77 EX の動作

Fortran 90 規格に基づき母国語型（日本語型）の引数をもつ組み込み関数(LEN, INDEX)が文法で正式に定義された。それぞれの日本語文字はすべて1文字として数えられる。

- ・非互換の検出方法

I レベルの診断メッセージが出力される。

- ・非互換の回避方法

FORTRAN 77 EX コンパイラで翻訳し直し、エラーメッセージが出力された非互換行を正しい文字長単位に修正する。

(17) IBSET, IBCLR, BTEST 関数の第2引数に範囲外の値を指定したとき動作が異なる。

- ・FORTRAN 77 の動作

なにもせず処理を続行する。

- ・FORTRAN 77 EX の動作

JWE 0257I-S メッセージを出力し、処理を続行する。

- ・非互換の検出方法

JWE 0257I-S メッセージが出力される。

- ・非互換の回避方法

第2引数の値が範囲を越えないように修正する。

(18) PARAMETER 文の定数式の中に論理型と算術型が混在するか、または PARAMETER 文の左辺と右辺が算術型と論理型の組合せであるとき、動作が異なる。

- ・ FORTRAN 77 の動作
JZK 375-W メッセージが出力され、実行される。
- ・ FORTRAN 77 EX の動作
翻訳時に S レベルのエラーメッセージが出力される。
- ・ 非互換の検出方法
翻訳時に S レベルのエラーメッセージが出力される。
- ・ 非互換の回避方法
論理値を算術型で演算しないよう、論理値を算術型に修正する。

(19) FORTRAN GE/HE で作成したオブジェクトと結合編集時にリンクされる実行時ライブラリに関する非互換。

- ・ FORTRAN 77 の動作
リンク可能である。
- ・ FORTRAN 77 EX の動作
リンクできない。
- ・ 非互換の検出方法
一般的な検出方法はない。
- ・ 非互換の回避方法
FORTRAN 77 EX コンパイラで再翻訳する。

5. Fortran 90 の概要

Fortran 77 言語仕様に対して追加された主な機能を文献[2]にしたがって簡単に紹介する。規格化の歴史、動向および日本の寄与等については[2,3]を参照されたい。また、最近出版された Fortran 90 の入門書として、ANSI および ISO のメンバーによって執筆された[10]がある。

(1) 自由形式によるソースプログラム記述の標準化

- ・ 行の最後に&を置き、次の行に継続できる。
- ・ コメントは "!" に続けて記述（行の先頭でも最後でもよい）。

(2) 配列操作機能の拡張

- ・ 配列の全体または部分についての代入や演算を一括して記述できる。
- ・ 配列を実行時に動的に割り付けることができる。

(例)

```
REAL, DIMENSION(100)::A, B, C, D      ! 実数型配列宣言。型と属性は", "で区切る
INTEGER, DIMENSION (:), ALLOCATABLE :: LD  ! 整数型の動的配列を定義する
INTEGER :: N
.....
D=A+B*2.0+SIN(C)                       ! 配列要素ごとの演算と代入
PRINT *, D
READ *, N
```

ALLOCATE(LD(N))

！実行時に配列を動的に確保する

(3) ポインタ変数(Pointer Variables)

- ・ポインタ機能が Fortran にも追加された。しかし、Fortran のポインタはC言語の「メモリ番地」としてのポインタとはだいぶ異なり、イメージ的にはむしろデータの単なる別名(alias)あるいは一般化された記述子(descriptors)と考えた方がよい。

(例)

```

REAL :: X, Y                                ! 実数型変数 X, Y の定義
REAL, DIMENSION (8,8) :: RX                 ! 実数型 2次元配列 RX の定義
REAL, POINTER :: PX, PY                     ! 実数型ポインタ変数 PX, PY の定義
REAL, DIMENSION (8,8), POINTER :: PRX       ! 配列 RX と同じ型のポインタ変数の定義
.....
X = 4.0 ; Y = 8.0                           ! ";" で区切るにより 1 行中に複数の文が書ける
PX => X ; PY => Y                           ! PX, PY をそれぞれ X, Y のポインタとする
PRINT *, PX                                ! PRINT *, X と全く同じ結果になる (4.0 が出力)
PY = PX                                    ! Y = X を実行したのと同じ結果になる (X = Y = 4.0)
PRX => RX(4,:)                             ! PRX は配列 RX 要素のうち第4行のポインタとなる
PRX = 0.0                                  ! この結果, RX(4,1)=RX(4,2)=...=RX(4,8)=0.0 となる

```

(4) 精度指定機能

- ・精度の指定 (KIND指定子) および精度の問い合わせが可能。

(5) 多種類の文字型 (各国語操作機能)

- ・従来の文字型のほかに、処理系の定める文字型データ (日本語や中国語などの複数文字データ) を同時に処理できる。
- ・文字型は種類 (KIND指定子) で区別する。KIND指定がないときは従来の文字型。

(例)

```

CHARACTER(KIND=2, LEN=11) :: A ! 長さが11の第2種文字型宣言
A=2_ '日本ソフトウェア科学会' ! 第2種文字型定数 (日本語) の代入

```

(6) 利用者定義のデータ型 (派生型: Derived Types)

- ・データ構造とそのデータに対する演算を利用者が定義できる。

(7) モジュール(Modules)

- ・大域的なデータおよびそのデータを操作する手続きを宣言する手段。
- ・派生型と組み合わせることによりデータ抽象化を実現できる。

(8) その他

- ・英小文字が使える。
- ・31文字以下の名標 (変数名や定数名など) が使える。
- ・1行中に ";" で区切って複数の文が書ける。
- ・別のファイルからソースプログラムを取り込む INCLUDE 機能。
- ・データ型のデフォルトを無効にする IMPLICIT NONE 機能。

- ・選択的分岐のための CASE 文。
- ・END DO や DO WHILE などの新しい DO 構造。
- ・NAMELIST によるネームリスト入出力のための英字名の宣言。
- ・内部手続き。
- ・手続きの再帰呼出し (RECURSIVE キーワード) 。
- ・手続きのインタフェースを定義するためのインタフェースブロック。

(9) Obsolescent (時代遅れ) な仕様

以下の仕様は、従来の Fortran 77 と互換性を保つため Fortran 90 でも一応残されているが、プログラミング言語仕様としてはすでに時代遅れであり、将来規格から削除される可能性が大きい。新規にプログラムを書く場合は以下の仕様はできるだけ避けるのが望ましい。

- ・算術 IF文
- ・実数型または倍精度型の DO制御変数およびループ制御式
- ・DO文末の共用
- ・DO文末が CONTINUE文や END DO文でない
- ・IF ブロックの外側から ENDIF文への分岐
- ・選択戻り指定子
- ・PAUSE文
- ・ASSIGN文、割当て型 GO TO文
- ・ASSIGN文による FORMAT指定

6. おわりに

この記事が読者に届く頃 (9月下旬) にはまだ FORTRAN 77 EX および FORTRAN 77 EX/VP は公開されていない。現在センター内で運用方式の検討やライブラリ等の非互換チェックを行なっている段階であり、残念ながら本稿ではTSSやバッチから新コンパイラを起動するための具体的な方法については述べるができなかった。遅くとも11月上旬までには旧コンパイラと新コンパイラの併設運用を開始する予定である。数カ月間の併設運用後、新コンパイラをセンターの標準 FORTRAN コンパイラとし、さらに将来的には旧コンパイラを完全に廃止する予定である。併設運用後は、できるだけ早い時期に既存プログラムの互換性チェックを行い、新コンパイラに移行していただきたい。具体的な使用方法 (起動コマンドやジョブ制御文など) および新旧コンパイラの運用形態についてはセンターニュースで広報するので注意されたい。

なお、京都大学大型計算機センターでは1990年10月1日から FORTRAN 77 EX のテスト運用を開始し、1991年4月1日から正式運用が始まった[11,12]。また、名古屋大学大型計算機センターでも1991年6月27日から新コンパイラが使えるようになった[13]。当センターの利用資格をもつ利用者は、SINSEI コマンドによりこれらのセンターに第二センター登録を行ない、N1ネットワーク経由でこれらの新コンパイラを使うことができる。SINSEI コマンドおよびN1ネットワークの使用法については[14]を参照されたい。

参考文献

- [1] 高木利久, FORTRANのアメリカ規格改訂について,
九州大学大型計算機センター広報, Vol.11, No.1, 1978, pp.10-16.
- [2] 和田英穂, Fortran 90の概要,
コンピュータソフトウェア, Vol.7, No.3, July 1990, pp.291-294.
- [3] 菅 忠義, FORTRAN 8Xの紹介, bit, Vol.17, No.12, pp.8-17, 1985.
- [4] 全国共同利用大型計算機センター運用会議計算機言語研究会,
全国共同利用大型計算機プログラミング言語の比較
FORTRAN -JIS FORTRAN(上位水準)を基準として- 第4版, 昭和59年1月.
- [5] 平野, SSU配列機能を用いた大規模ベクトル演算について,
京都大学大型計算機センター広報, Vol.24, No.3, 1991, pp.145-149.
- [6] 計算機マニュアル, FORTRAN 77 文法書 1991年6月版(99SP-8032-1), 富士通㈱, 1990.
- [7] 計算機マニュアル, OSIV/MSP FORTRAN 77 EX V12用 使用手引書
(79SP-5031-1), 富士通㈱, 1991.
- [8] 計算機マニュアル, OSIV/MSP FORTRAN 77 EX/VP 使用手引書 V12用
(79SP-5041-1), 富士通㈱, 1991.
- [9] 計算機マニュアル, OSIV/MSP FORTRAN 77 EX メッセージ説明書 V12用
(70SP-5321-1), 富士通㈱, 1991.
- [10] W.S. Brainerd, C.H. Goldberg, and J.C. Adams,
Programmer's Guide to Fortran 90, McGraw-Hill, 1990.
- [11] FORTRAN 77-EXのテスト運用について,
京都大学大型計算機センターニュース, No.9, 1990, pp.53.
- [12] FORTRANコンパイラについて,
京都大学大型計算機センターニュース, No.3, 1991, pp.20-22.
- [13] Fortranコンパイラが変わります,
名古屋大学大型計算機センター速報, No.277, 1991, pp.1-5.
- [14] 利用の手引 ネットワーク (第三版), 九州大学大型計算機センター, 1990.