

第10章 非同期入出力に関する公認拡張

本章では、HPF プログラムまたは Fortran プログラムから、非同期入出力を実行するための機構を定義する。これは、Fortran95 規格草案 X3J3/96-007r1 への変更として示される。この拡張は、X3J3 非同期入出力拡張提案である論文 X3J3/96-158r2 の一部である。一言でいえば、この拡張は、書式無し直接データ転送を、プログラムの実行とは非同期的に実行できるようにするためのものである。データ転送の完了を待ち合わせるためには、WAIT 文を使用することができる。データ転送が完了したかどうかを判定するためには、INQUIRE 文を使用することができる。

第 9.3.4 項の規則 R905 *connect-spec* に、以下を追加する。

or ASYNCHRONOUS

第 9.3.4.10 項の後に、「OPEN 文の ASYNCHRONOUS 指定子」というタイトルの新しい項を追加する。これは、以下のような段落を含む。

OPEN 文中の装置に対して ASYNCHRONOUS 指定子が指定されている場合、その装置に対する READ 文及び WRITE 文の制御情報並び中に、ASYNCHRONOUS 指定子を指定してもよい。

READ 文又は WRITE 文中に ASYNCHRONOUS 指定子が指定された場合、プロセッサは、データ転送を非同期的に実行してもよい(ただし、必ずしもその必要はない)。WAIT 文は、非同期入出力処理を待ち合わせるため、又は非同期入出力処理の状態を問い合わせるために用いられる。

第 9.4.1 項の規則 R912 *io-control-spec* に以下を追加する。

or ID = *scalar-default-int-variable*
or ASYNCHRONOUS

さらに、以下の制約を追加する。

制約: ASYNCHRONOUS 指定子又は ID=指定子のどちらか一方を指定した場合、その両方を指定しなければならない。

制約: ASYNCHRONOUS 指定子を指定した場合、REC=指定子を指定しなげばならず、かつ「書式」を指定してはならず、かつ「変数群名」を指定してはならない。

制約: ASYNCHRONOUS 指定子を指定した場合、データ転送文中のどの式においても、関数を引用してはならない。

第 9.4.1 項の終わりに、以下の段落を追加する。

ID=指定子を追加することにより、非同期データ転送が開始される。データ転送文が ID 変数に返す値と同じ値の ID が指定された WAIT 文が、データ転送文の実行後、最終的に実行されなければならない。この WAIT 文を、「対応する」WAIT 文と呼ぶ。非同期データ転送は直接かつ書式無しでなければならないことに注意されたい。

対応する WAIT 文は、非同期データ転送文が実行されたのと同じ副プログラムの同一の分身 (instance) で実行されなければならない。

【実装者への助言】 この制約は、コンパイラによる、WAIT 文をまたぐコード移動最適化を防ぐためのものである。非同期入出力並び中に指定された変数に関連するいかなる処理も、対応する WAIT 文が実行されるまでに行ってはならない。【以上】

io-unit を開く時に、ASYNCHRONOUS 指定子が省略された場合、ASYNCHRONOUS 指定子も ID=指定子も指定してはならない。

第 9.4.1 項の制約の後の、4 番目と 5 番目の段落中に現れる「IOSTAT=指定子、又は SIZE=指定子」を、「IOSTAT=指定子、SIZE=指定子、又は ID=指定子」に変更する。

以下の文章を、第 9.4.3 項の最後の段落の前に挿入する。

非同期データ転送の場合、誤り条件はデータ転送文の実行中か、後続するデータ転送の間に発生する。誤り条件がデータ転送文中で発生し、その結果、プログラムが終了しなかった場合、誤り条件はデータ転送文中の ERR=指定子と IOSTAT=指定子によって検出できる。誤り条件が後続するデータ転送の間に発生し、その結果、プログラムが終了しなかった場合、誤り条件は、対応する WAIT 文中の ERR=指定子と IOSTAT=指定子によって検出できる。

第 9.4.3 項の最後の段落の、「実行」の最初の出現を、「実行と後続するデータ転送」に変更する。

第 9.4.4 項に、以下の文章を追加する。

非同期データ転送では、手順 1-8 は、非同期データ転送文と対応する WAIT 文の両方に対応する。手順 4-7 は、プログラムの実行とは非同期的に実行される可能性がある。非同期データ転送をサポートしない実装では、手順 1-8 は、非同期データ転送文によって実行される。この場合でも、対応する WAIT 文は、状態情報を返すことがただ一つの効果であるように実行されなければならない。

非同期データ転送文中に、「入力項目」又は「出力項目」として指定された任意の変数、又はそのような変数と結合している任意の変数は、対応する WAIT 文が実行されるまで、参照したり、定義状態又は不定状態にしてはならない。ただし、ポインタは、そのような変数と結合してもよい。

未完了の非同期データ転送処理 (READ 文又は WRITE 文) が複数あってもよい。ただし、2 つの WRITE 処理に対して、間に WAIT 文をはさむことなく同一の装置番号や記録番号を指定してはならない。

1 【利用者への助言】 HPF では、非同期であるかどうかに関わらず、READ 文の入出力並
2 び中で、左の項目が右の項目を定義してもよい。これは、次のような文、

3
4 READ(10, ID=IDNUM, REC=10) I, A(I)

5
6
7 が規格合致であり、同期 READ 文と同じ入力動作となることを意味する。【以上】

8
9 第 9 章では、「及び INQUIRE 文」を、「INQUIRE 文、及び WAIT 文」に変更する。
10 第 9.6.1.14 項の段落の最後の文として、以下の文を追加する。

11
12 指定された装置に対して未完了のデータ転送処理がある場合、NEXTREC=指定子に
13 代入される値は、全ての未完了のデータ転送が既に完了しているかのように、そ
14 の発行順に計算される。

15
16 第 9.6.1 項の規則 R924 *inquire-spec* に、以下を追加する。

17
18 or ID = *scalar-default-int-variable*
19 or PENDING = *scalar-default-logical-variable*

20
21 さらに、以下の制約を追加する。

22 制約: INQUIRE 文中に FILE=指定子を指定した場合、ID=指定子や PENDING=指定
23 子を指定してはならない。

24 制約: ID=指定子又は PENDING=指定子のどちらか一方を指定する場合、その両方
25 を指定しなければならない。

26
27
28 第 9.6.1.22 項の後に、「INQUIRE 文の ID=指定子と PENDING=指定子」というタ
29 イトルの新しい項を追加する。これは、以下のような段落を含む。

30
31 INQUIRE 文中に ID=指定子が指定された場合、PENDING=指定子中に指定された変
32 数の値が真になるのは、ID=指定子で識別される、指定された装置に対するデー
33 タ転送が、未完了の場合だけである。それ以外の場合には、PENDING=指定子中に
34 指定された変数の値は偽になる。

35 36 10.1 WAIT 文

37
38
39
40 H1001 *wait-stmt* is WAIT (*wait-spec-list*)

41 H1002 *wait-spec* is UNIT = *io-unit*
42 or ID = *scalar-default-int-expr*
43 or ERR = *label*
44 or IOSTAT = *label*
45

46
47 制約: *wait-spec-list* には、UNIT=指定子をちょうど 1 つ、ID=指定子をちょうど 1 つと、他の
48 各指定子を高々 1 つ、指定しなければならない。

WAIT 文により、非同期データ転送が完了する。IOSTAT=指定子と ERR=指定子は省略可能
であり、それらは、それぞれ第 9.4.1.4 項と第 9.4.1.5 項で説明されている。

【実装者への助言】 実装者は、任意の又は全ての非同期入出力を同期的に実行しても
よい。このような場合は本質的に、データ転送文の転送結果を記録するために ID=節を
使用し、その結果を対応する WAIT 文に渡すことを意味している。【以上】

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48