



Title	FORTRAN 77概説 (2)
Author(s)	大中, 幸三郎; 後藤, 米子
Citation	大阪大学大型計算機センターニュース. 1982, 44, p. 123-141
Version Type	VoR
URL	https://hdl.handle.net/11094/65520
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

FORTRAN 77 概説 (2)

大阪大学大型計算機センター研究開発部

大 中 幸三郎・後 藤 米 子

11. 入 出 力 文

11.1 記 錄

記録には、書式付き記録、書式なし記録およびファイル終了記録の3つがある。

書式付き記録は文字の列で構成され、その長さは文字の個数(バイト数)によって数える。書式なし記録は、内部記憶における表現のビットの列によって構成され、その長さは語を単位とする次式で与えられる。

$$\text{書式なし記録の長さ} = \sum_i n_i + \sum_j \text{INT}((k_j + 3) / 4)$$

ただし、 n_i は記録中の連続する文字型以外の要素の i 番目の集まりの語数、 k_j は記録中の連続する文字型の要素の j 番目の集まりの文字数とする。

ファイル終了記録はENDFILE文および順呼び出し出力文に続くCLOSE文、BACKSPACE文、REWIND文によって書かれる。FORTRAN77にのみ存在するCLOSE文(11.6参照)を除けば、ACOS FORTRAN66と77で差異はないが、JIS FORTRAN77ではBACKSPACE文とREWIND文にファイル終了記録を書く機能はない。

11.2 フ ァ イ ル

11.2.1 外部ファイルと内部ファイル

ファイルには外部ファイルと内部ファイルの2つがある。

外部ファイルは、従来、単にファイルと呼ばれていたものである。一つのファイル中に書式付きと書式なしの記録を混在させることができないこと以外は、ACOS FORTRAN66と77で差異はない。

内部ファイルは、READ文およびWRITE文を使って、内部記憶間における変換を伴ったデータ転送を可能とするものである。ACOS FORTRAN66におけるENCODE文およびDECODE文の機能にはほぼ等しい。内部ファイルの性質を次に示す。

- 内部ファイルは、文字型であり、変数、配列要素、配列または部分列のいずれかである。
- 内部ファイルの一つの記録は、一つの変数、配列要素または部分列である。すなわち、内部ファイルが配列以外の場合は、そのファイルには一つの記録しかない。配列の場合には、各配列要素が一つの記録となる。

- 出力される文字数が記録の長さより短い場合には、残りは空白となる。記録の長さより長い文字数を出力することはできない。
- 内部ファイルに対する入出力文として使用できるものは、並びによる書式（暗黙書式）を除く順番探査書式付きのREAD文とWRITE文のみである。
- データ転送の前には、常に最初の記録の先頭に位置づけられる。

11.2.2 外部ファイルの探査と編成

外部ファイル中の記録を探査する方法には、順番探査（順呼び出し）と直接探査（直接呼び出し）がある。順番探査は順編成ファイルに対して用いられ、直接探査は直接編成ファイルに対して用いられる。

ACOS FORTRAN66と77において使用されるファイルアクセスシステムと標準のファイル形式を次表に示す。

	ファイルアクセス システム ^{注)}	ファイル形式	
		順編成	直接編成
FORTRAN 77	UFAS	FRC形式	UFF形式（相対編成）
FORTRAN 66 (R)	FRC	FRC形式	FRC形式
FORTRAN 66 (V)	UFAS(V)	FRC形式	UFF形式（相対編成）

注) UFAS ユニファイドファイルアクセスシステム
FRC ファイル / レコードコントロール

11.2.3 ファイルの存在

ファイルの存在には、物理的な存在と論理的な存在が考えられる。物理的な存在とは、ファイルの実体が存在することであり、論理的な存在とはFORTRANプログラムから使用可能なことを言う。すなわち、プログラムを実行させている課題番号と同一の課題番号の下のファイルに対して、物理的な存在と論理的な存在は一致するが、異なる課題番号の下のファイルはパーミッションに依存し、物理的に存在しても論理的に存在しないことがある。

JIS FORTRAN77でのファイルの存在は、論理的な存在のことを意味するが、ACOS FORTRAN77でのファイルの存在は、マニュアルでは必ずしも統一されていないが、物理的な存在を意味する。

11.3 接続と事前の接続

接続とはファイルを参照するための装置とファイルとの関係であり、次の条件を満足するときのことである。

- (1) 装置とファイルの間に1対1の対応関係が決定している。
- (2) 探査方法が決定している。
- (3) 書式の有無が決定している。

- (4) 直接探査の場合には、記録の長さが決定している。
- (5) 書式付きの場合には、数値入力欄中の空白の取扱いが決定している。(11.5.2(9)参照)
- 事前の接続とは、実行可能プログラムの実行に先立ち、ジョブ制御言語あるいはTSSのコマンドによって接続の5つの条件の中の一部が満足されるときをいう。必ず満足しなければならない条件は、ACOS FORTRAN77とJIS FORTRAN77では次表のように異なる。

	ACOS	JIS
満足される条件	(1), (5) ^注	(1), (2), (4), (5) ^注
必ずしも満足されなくてよい条件	(2), (4)	(3)
満足されない条件	(3)	ナシ

注) 空白の取扱いは、ACOSではNULLとなるが、JISではNULLまたはZEROのいずれかが決定していればよい。NULLとZEROの意味については、11.5.2(9)を参照のこと。

ACOS FORTRAN77では、装置を識別する番号5～7、41～43も次表に示すように、事前の接続がなされている。

	ACOS FORTRAN77	ACOS FORTRAN66
カード読み取り装置、端末	5, 41	5, 41
ラインプリンタ、端末	6, 42	6, 42
カードせん孔装置	7, 43	43

事前の接続において満足されない条件を決定し、接続することは、ACOS FORTRAN77ではOPEN文あるいは最初のREAD文、WRITE文によって行われるが、JIS FORTRAN77では最初のWRITE文によってのみ行われる。このちがいは事前の接続の内容の差に依存しているが、OPEN文、CLOSE文およびINQUIRE文においても、同じ理由によって、ACOSとJISの間に多少の差異がある。また、ACOS FORTRAN77では、事前の接続と接続の二つの状態を合せて「割当て」と呼んでいるが、JIS FORTRAN77には割当てという考えはない。

11.4 READ文およびWRITE文

11.4.1 形 式

READ文とWRITE文は次の形である。

READ(*cl*) *iol*

ただし、*cl* は制御情報並び、*iol* は入出力並びとする。

WRITE(*cl*) *iol*

この形はFORTRAN66と同じであるが、*cl* と *iol* の内容は少し異っている。

11.4.2 制御情報並び

(1) UNIT = *u*

外部装置または内部ファイルを指定するものであり、省略できない。*u* は次のいずれかである。

- 1～63の値をもつ整数型の式。

- *印。(READ文では5、WRITE文では6の指定と等価。)
 - 内部ファイルを示す文字型の変数、配列要素、配列あるいは部分列。
- } FORTTRAN 66
では不可。

JIS FORTRAN 77では、整数型の式の値はゼロまたは正となっており、上記よりは少し広くなっている。ACOS FORTRAN 66ではRモードで1～43、Vモードで1～63であった。

UNIT=を省略して、単に*u*と書くこともできるが、この場合は制御情報並びの最初の要素でなければならない。

(2) FMT = *f*

書式を指定するものであり、*f*は次のいずれかである。

- FORMAT文を示す文番号またはスイッチ変数。
- 文字型の配列名。
- 文字式。ただし、擬文字長指定または整合文字長指定を受けたオペランド(定数名は除く)を連結した文字式は許されない。
- *印。並びによる書式(暗黙書式、12.5参照)を示す。

FMT=を省略して単に*f*と書くこともできるが、この場合は制御情報並びの最初の要素は*u*(UNIT=*u*は不可)、二番目の要素は*f*でなければならない。

(3) REC = *rn*

直接探査の場合に、入出力しようとする記録番号を指定する。*rn*は整数型の式でその値は正でなければならない。ACOS FORTRAN 66と差異はない。

(4) ERR = *s₁*, END = *s₂*

ERR = *s₁*は、入出力文の実行中に誤りが発生した場合(以下誤り条件と言う)の処理を指定し、END = *s₂*は入力文の実行中にファイル終了条件が発生した場合の処理を指定する。*s₁*および*s₂*は文番号またはスイッチ変数であり、ACOS FORTRAN 66と同じであるが、JIS FORTRAN 77ではスイッチ変数は許されていない。

ACOS FORTRAN 77における誤り条件とは、FORTRANとUFAS(Unified File Access System)の実行時のエラーが発生する場合である。ファイル終了条件とは、外部ファイルに対してはファイル終了記録を検出した場合であり、内部ファイルに対しては最後の記録を越えて入力しようとした場合である。

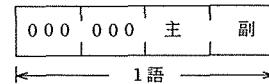
(5) IOSTAT = *ios*

入出力文の実行の状態を示すものであり、整数型の変数または配列要素 *ios* は次の値によって確定となる。

- 誤り条件もファイル終了条件も発生しなかった場合、ゼロ。

- 誤り条件が発生した場合、正。ただし、FORTRAN

の実行時エラーの場合はエラー番号が入り、UFAS
の場合は主、副のステータスが右図のように入る。



- ファイル終了条件が発生した場合、-1。

誤り条件あるいはファイル終了条件が発生したにもかかわらず、ERR= s_1 あるいはEND= s_2 の指定がない場合の実行は、IOSTAT= ios を含んでいれば入出力文の次の実行文となり、含んでいなければ終了する。

11.4.3 入出力並び

下記に注意をする必要がある以外は、ACOS FORTRAN 66と同じである。

- 入出力並びに、文字部分列名を指定できる。
- 入出力並びに、擬寸法配列名は指定できない。
- 出力並びに、入出力文の実行を引き起す関数を含む式は指定できない。
- 出力並びに、擬文字長指定、または整合文字長指定を受けたオペランドの連結を含む式は、指定できない。ただし、このオペランドが定数名であれば許される。
- DO 形並びの処理は、DOループの場合（10.9参照）と全く同じである。

11.4.4 省略形

下表の省略形も許されている。ただし、×印は許されていないものを示す。

	JIS FORTRAN 77	ACOS FORTRAN 77	ACOS FORTRAN 66
READ f, iol	READ(*, f) iol	READ(5, f) iol	READ(41, f) iol
READ, iol	×	READ(5, *) iol	READ(41, *) iol
PRINT f, iol	WRITE(*, f) iol	WRITE(6, f) iol	WRITE(42, f) iol
PRINT, iol	×	WRITE(6, *) iol	WRITE(42, *) iol
PUNCH f, iol	×	WRITE(7, f) iol	WRITE(43, f) iol
PUNCH, iol	×	WRITE(7, *) iol	WRITE(43, *) iol

11.5 OPEN文

11.5.1 目的と形式

OPEN文はFORTRAN 77で新しく追加された機能であり、プログラムの実行中にファイルと装置を接続し、入出力を可能とする文である。FORTRAN 66では、プログラムの実行の前にジョブ制御言語あるいはTSSのコマンドを用いることしかできなかった。

OPEN文は次の形である。

OPEN(ol) ただし、 ol は指定子の並びとする。

11.5.2 指 定 子

(1) UNIT = *u*

u が 1 ~ 63 の値をもつ整数型の式に制限される以外は、READ文およびWRITE文の場合と同一である（11.4.2(1)参照）。この指定子は必ず必要である。

(2) ERR = *s₁* , IOSTAT = *ios*

11.4.2(4), (5)参照。

(3) FILE = *fin*

装置と接続されるファイルの名前を指定する。*fin* は文字式であり、後に続く空白を取り除いた値が、下記のファイル記述子となっていなければならない。

課題番号 / *c₁* \$ *p₁* / ··· / *c_n* \$ *p_n* / ファイル名 \$ *p_f*

ただし、*c_i* はカタログ名、*p_i* と *p_f* はパスワードを示す。

(4) STATUS = *sta*

sta は文字式であり、後に続く空白を取り除いた値は、OLD, NEW, SCRATCH または UNKNOWN のいずれかである。この指定子の効果を次に示す。

- OLD FILE 指定子があり、そのファイルが論理的に存在する場合に、ファイルと装置を接続する。他の場合はエラーとなる。
- NEW FILE 指定子があり、そのファイルが物理的に存在せず、ファイルを生成するパーミッションが付いている場合に、ファイルを生成し、装置と接続する。他の場合はエラーとなる。
- SCRATCH FILE 指定子がない場合に、テンポラリファイルと装置を接続する。
FILE 指定子がある場合はエラーとなる。
- UNKNOWN パーミッションによって拒否される場合にエラーとなる以外は、それぞれの場合に応じて、OLD, NEWあるいはSCRATCHと等価となる。
ただし、すでに割当てられているときには全く等価ではない（11.5.3 参照）。

この指定子を省略すると、UNKNOWNを指定したものとみなされる。

(5) ACCESS = *acc*

ファイルの探査方法を指定する。*acc* は文字式であり、後に続く空白を取り除いた値は、SEQUENTIAL または DIRECT のいずれかである。この指定子を省略すると、SEQUENTIAL を指定したものとみなされる。

(6) FORM = *fm*

ファイルを書式付き入出力または書式なし入出力で接続することを指定する。*fm* は文字式であり、後に続く空白を取り除いた値は、FORMATTED または UNFORMATTED のいずれかであ

る。この指定子を省略すると、順番探査の場合はFORMATTED、直接探査の場合はUNFORMATTEDを指定したものとみなされる。

(7) RECL = *rl*

直接探査の場合に、一記録の長さ（11.1参照）を指定する。*rl*は整数型の式であり、その値は正でなければならない。この指定子は、直接探査の場合は省略できず、順番探査の場合は指定してはならない。

(8) MAXREC = *maxr*

ファイル中に含み得る記録の数を指定する。*maxr*は整数型の式であり、その値は正でなければならない。*acc*の値がDIRECTかつファイルを生成するときにのみ指定することができる。この指定子を省略すると、1リンク（3840語）の大きさのファイルが生成される。JIS FORTRAN 77には存在しない指定子である。

(9) BLANK = *blk*

書式付き入力において、数値入力欄中の空白の取扱いを指定する。*blk*は文字式であり、後に続く空白を取り除いた値はNULLまたはZEROでなければならない。NULLの場合には、空白はすべて無視される。ただし、すべてが空白である欄はゼロとみなされる。ZEROの場合には、先行する以外の空白はゼロとみなされる。この指定子は、*fm*の値がFORMATTEDのときにのみ指定することができる。省略された時にはNULLを指定したものとみなされる。したがって、FORTRAN 66との互換性を保つためには、必ずZEROを指定しなければならない。

11.5.3 実行上の注意

- すでに接続されているファイルと装置に対しては、BLANK指定子のみを変更してもよい。他の指定子(FILE指定子も含む)は省略するか、同じ値としなければならない。省略された場合には、11.5.2に示した省略時の値ではなく、接続されたときの値となる。ただし、FILE指定子を省略した場合には、STATUS指定子も省略するか、*sta*の値はUNKNOWNまたはSCRATCHでなければならない。
- ファイルがある装置とすでに割当てられているとき、その装置を異なるファイルと接続してはならない。
- 装置があるファイルとすでに割当てられているとき、その装置を異なるファイルと接続してもよい。この場合には、OPEN文の実行の直前にCLOSE文（11.6参照）が実行される。このCLOSE文のSTATUS指定子は省略したものとみなされ、IOSTAT指定子とERR指定子はOPEN文と同じものが指定されたものとみなされる。
- ACOS FORTRAN 77では、事前の接続の状態にあるファイルと装置に対して、決定していない指定子の値を定めるためにOPEN文を実行してもよい。すでに決定している指

定子は RECL 指定子と BLANK 指定子を除き、省略するか、事前の接続と同じ値としなければならない。省略された場合には、11.5.2 に示した省略時の値ではなく、事前の接続のときの値となる。ただし、FILE 指定子を省略した場合には、STATUS 指定子も省略するか、sta の値は UNKNOWN または SCRATCH でなければならない。RECL 指定子は直接探査のときには省略できず、BLANK 指定子は省略できるが、いずれの指定子も事前の接続時の値と異っていてもよい。この場合には OPEN 文の指定が有効となる。

- acc の値が SEQUENTIAL かつファイルを生成するときには、ACOS ではファイルの大きさは、初期値が 1 リンク、最大値が unlimited となる。acc の値にかかわらず、JIS FORTRAN 77 では、生成されるファイルの大きさについては記述されていない。

11.6 CLOSE 文

11.6.1 目的と形式

CLOSE 文は FORTRAN 77 で新しく追加された機能であり、プログラムの実行中にファイルと装置の接続を終了させるのに用いる。

CLOSE 文は次の形である。

CLOSE(*cell*) ただし、*cell* は指定子の並びとする。

11.6.2 指定子

(1) UNIT = *u*, IOSTAT = *iost*, ERR = *s₁*

11.5.2 (1), 11.4.2 (4), (5) 参照。

(2) STATUS = *sta*

u で指定した装置に接続されているファイルの後処理を指定する。*sta* は文字式であり、後に続く空白を取り除いた値は、KEEP または DELETE のいずれかである。JIS FORTRAN 77 では、KEEP を指定すると接続の終了後もファイルは存在し、DELETE を指定すると存在しなくなるが、ACOS FORTRAN 77 では、事前の接続に戻る場合がある（11.6.3 参照）。この指定子を省略すると、バッチ処理においてプログラムの実行中に生成されたテンポラリファイルに対して DELETE となる以外は、KEEP を指定したものとみなされる。

11.6.3 接続の終了

CLOSE 文の実行によって、接続は終了し、次の三つの状態のいずれかとなる。ただし、JIS FORTRAN 77 では、事前の接続に戻る場合はない。

- 事前の接続に戻る場合

バッチ処理では、事前の接続において、ディスポジションを S (セーブ) と指定されている磁気テープ装置あるいはテンポラリファイルに、KEEP を指定した場合である。TSS 処理では、事前の接続によるテンポラリファイルに、KEEP を指定した場合である。ただ

し、いずれの場合においても、記録の長さは接続されたときの値となっている。

◦ 割当てを解放するが、ファイルは存在する場合

名前付きファイルにKEEPを指定した場合である。

◦ 割当てを解放し、ファイルが存在しなくなる場合

上記の場合以外である。ディスポジションがSとなっている場合にも、DELETEが指定されると、ファイルが存在しなくなることに注意を要する。

プログラムの実行が終了するときにおける割当ての処理は、下記の場合を除いて、UNIT指定子のみを指定したCLOSE文の実行と同じとなる。ただし、JIS FORTRAN77では、どのような場合においても、UNIT指定子のみを指定したCLOSE文の実行と同じとなる。

- バッチ処理において、ディスポジションをSと指定されている磁気テープ装置あるいはテンポラリファイルは、割当てを解放されるが、ファイルは存在する。すなわち、ロジカルユニットデジグネータによって、次のアクティビティに引き継ぐことができる。
- TSS処理においては、名前付きファイルとテンポラリファイルのいずれの場合も、事前の接続に戻る。すなわち、AFT上に登録された状態となる。

11.6.4 実行上の注意

- バッチ処理において、プログラムの実行中に生成されたテンポラリファイルと接続されている装置に対して、KEEPを指定してはならない。JIS FORTRAN77では、処理形態によらず、KEEPを指定してはならないことになっている。
- 事前の接続の状態にあった名前付きファイルと接続されている装置に対して、DELETEを指定してはならない。この制限は JIS FORTRAN77 にはない。
- ファイルと接続されていない装置あるいは存在しない装置に対して CLOSE文を実行してもよい。

11.7 INQUIRE 文

11.7.1 目的と形式

INQUIRE文はFORTRAN77で新しく追加された機能であり、プログラムの実行中にファイルあるいは装置の状態を問合せるために用いる。

INQUIRE文は、ファイルと装置のいずれをキーとするかによって、ファイルによるINQUIRE文と装置によるINQUIRE文の二種類にわかれるが、形式は同一で次の通りである。

INQUIRE(*il*) ただし、*il* は指定子の並びである。

il の中にFILE指定子(11.5.2(3)参照)を含むものをファイルによるINQUIRE文と言い、UNIT指定子(11.5.2(1)参照)を含むものを装置によるINQUIRE文と言う。FILE指定子とUNIT指定子以外は問合せ指定子(11.7.2参照)を必要に応じて指定する。ただし、FILE指定

子とUNIT指定子の双方を *il* の中に指定してはならない。

11.7.2 問合せ指定子

ACOS FORTRAN 77 では、ファイルの存在と事前の接続の意味が、JIS FORTRAN 77 と異なっている（11.2.3, 11.3 参照）。したがって、問合せ指定子に対する返却値も異なる場合があるが、繁雑となるから、本節では ACOS と JIS の差異については述べない。

(1) $\text{ERR} = s_1$, $\text{IOSTAT} = ios$

11.4.2 (4), (5) 参照。

(2) $\text{EXIST} = ex$

ex は論理型の変数または配列要素である。ファイルによる INQUIRE 文では、ファイルが物理的に存在する場合は真、物理的に存在しない場合は偽となる。装置による INQUIRE 文では、UNIT指定子の *u* の値が 1～63 のときに真、1～63 以外のときに偽となる。

(3) $\text{OPENED} = od$, $\text{NUMBER} = num$

od は論理型、*num* は整数型の変数または配列要素である。ファイルと装置が割当てられている場合に *od* は真、*num* は装置の番号となり、割当てられていない場合に *od* は偽、*num* は不定となる。

(4) $\text{NAMED} = nmd$, $\text{NAME} = fn$

nmd は論理型、*fn* は文字型の変数または配列要素である。*nmd* と *fn* には次表の値が返却される。ただし、ファイルによる INQUIRE 文においてファイルが物理的に存在するときを (F)、ファイルが物理的に存在しないときを (f)、装置による INQUIRE 文のときを (U) で示す。

	割当てられているとき		割当てられていないとき		
	名前付きファイル	名前付きファイル以外	(F)	(f)	(U)
<i>nmd</i>	真	偽	真	偽	不定
<i>fn</i>	ファイル記述子 ^{注)}	不定	ファイル記述子	不定	不定

注) 事前の接続によって割当てられた場合にはファイル名となる。したがって、この値を FILE 指定子に与えてはならない。ただし、事前の接続において未定の指定子の値を OPEN 文によって定めたとき、FILE 指定子を省略しなければ、ファイル記述子となる。

(5) $\text{ACCESS} = acc$, $\text{SEQUENTIAL} = seq$, $\text{DIRECT} = dir$

acc, *seq* および *dir* はいずれも文字型の変数または配列要素である。*acc*, *seq* および *dir* には次表の値が返却される。ただし、ファイルによる INQUIRE 文においてファイルが物理的に存在するときを (F) で示す。

	割当てられているとき			割当てられていないとき	
	順番探査で接続	直接探査で接続	事前の接続	(F)	(F) 以外
<i>acc</i>	SEQUENTIAL	DIRECT	UNKNOWN ^{注)}	不定	不定
<i>seq</i>	YES	NO	UNKNOWN ^{注)}	UNKNOWN	不定
<i>dir</i>	NO	YES	UNKNOWN ^{注)}	UNKNOWN	不定

注) 磁気ディスク装置以外のときには、*acc*, *seq*, *dir* の値はそれぞれSEQUENTIAL, YES, NO となる。

(6) FORM = *fm*, FORMATTED = *fmt*, UNFORMATTED = *unf*

fm, *fmt* および *unf* はいずれも文字型の変数または配列要素である。*fm*, *fmt* および *unf* には次表の値が返却される ただし、ファイルによる INQUIRE 文においてファイルが物理的に存在するときを (F) で示す。

	割当てられているとき			割当てられていないとき	
	書式付きで接続	書式なしで接続	事前の接続	(F)	(F) 以外
<i>fm</i>	FORMATTED	UNFORMATTED	UNKNOWN ^{注)}	不定	不定
<i>fmt</i>	YES	NO	UNKNOWN ^{注)}	UNKNOWN	不定
<i>unf</i>	NO	YES	UNKNOWN ^{注)}	UNKNOWN	不定

注) カード読取装置、ラインプリンタ、カードせん孔装置および端末のときには、*fm*, *fmt*, *unf* の値はそれぞれ FORMATTED, YES, NO となる。

(7) RECL = *rcl*, NEXTREC = *nr*

rcl および *nr* はいずれも整数型の変数または配列要素である。*rcl* および *nr* は直接探査によってファイルと装置が割当てられている場合にのみ下記の値となり、他の場合は不定となる。

rcl 一記録の長さ。ただし、単位は FORM 指定子の値が UNFORMATTED のときには語数、他のときには文字数となる(11.1参照)。

nr 最後に参照された記録の記録番号 + 1。ただし、一度も参照されていないときには 1 となる。

(8) BLANK = *blk*

blk は文字型の変数または配列要素である。*blk* はファイルと装置が割当てられている場合に下記の値となり、割当てられていない場合は不定となる。

blk { 書式付き入出力で接続 OPEN 文の BLANK 指定子の値 (NULL または ZERO)。
 書式なし入出力で接続 不定。
 事前の接続 NULL。

11.8 BACKSPACE 文、ENDFILE 文およびREWIND 文

BACKSPACE 文、ENDFILE 文およびREWIND 文は次の形である。

BACKSPACE(*al*) ただし、*al* はUNIT指定子、 IOSTAT指定子、 ERR 指定
ENDFILE(*al*) 子の並びであるが、 UNIT指定子を必ず含んでいなければなら
REWIND(*al*) ない。各指定子については 11.5.2(1), 11.4.2(4), (5)を参照のこと。

FORTRAN 66 同じ形の BACKSPACE *u*, ENDFILE *u*, REWIND *u* も許されるが、これらは、*al* にUNIT指定子のみを指定した場合に等しい。

機能としては、ACOS FORTRAN 66と77の間に差はないが、JIS FORTRAN 77では BACKSPACE文とREWIND文によってファイル終了記録を書くことはできない(11.1参照)。

12. 書式仕様

12.1 書式仕様の種類

書式は、次のいずれかによって与える。

- (1) FORMAT文。
- (2) 文字型の変数、配列要素あるいは配列の値。
- (3) 文字式の値(文字定数を含む)。FORTRAN 66 では不可。
- (4) *印。並びによる書式(暗黙書式、12.5 参照)を示す。

12.2 書式仕様の形式

12.2.1 編集記述子

編集記述子(欄記述子)には反復可能編集記述子と反復不能編集記述子の二種類がある。

$$\begin{aligned} \text{反復可能編集記述子} & \left\{ \begin{array}{l} Iw, Iw.m, Fw.d, Ew.d, Ew.dEe, Dw.d, Qw.d, \\ Gw, Gw.d, Gw.dEe, Lw, A, Aw, Rw. \end{array} \right. \\ \text{反復不能編集記述子} & \left\{ \begin{array}{l} 'h_1 h_2 \cdots h_n', "h_1 h_2 \cdots h_n", nHh_1 h_2 \cdots h_n, \\ Tc, TLc, TRc, nX, /, :, S, SP, SS, kP, BZ, BN. \end{array} \right. \end{aligned}$$

ただし、*w*, *e*, *n*, *c* はゼロでない符号なし整定数、*d*, *m* は符号なし整定数、*k* は整定数、*h_j* は JIS 文字である。

ACOS FORTRAN 66になかったものは、Iw.m, Ew.dEe, Gw.dEe, TLc, TRc, :, S, SP, SS, BZ, BN である。また、/ と kP は欄区切りと桁移動子と呼ばれていたものであるが、編集記述子として扱われるために、コンマで区切って用いることができるようになった。ただし、編集記述子の間のコンマは次の場合には省略でき、FORTRAN 66との互換性を考慮している。

- (1) P 形編集記述子と、その直後の F, E, D, Q あるいは G 形編集記述子の間。
- (2) 斜線編集記述子の前後。
- (3) コロン編集記述子の前後。

(4) X, H, アポストロフィおよび引用符編集記述子の直後。ただし、連続するアポストロフィ編集記述子の間、および連続する引用符編集記述子の間のコンマは省略できない。

Gw , Rw , " $h_1 h_2 \cdots h_n$ " およびコンマの省略の(4)は JIS FORTRAN 77 では許されてはいない。

12.2.2 行送り文字

ACOS FORTRAN 77 で使用できる行送り文字は 0 ~ 9、空白、+ および & の 13 種類であり、ACOS FORTRAN 66 に & が追加された。JIS FORTRAN 77 では 0、1、空白および + の 4 種類のみが許されている。

行送り文字 & は TSS 処理の場合にのみ有効であり、行送りをせずに直前の行に継続して印字される。

例 WRITE(6, 1)
1 FORMAT(1H_, 'ABC', /, 1H&, 'XYZ') } \Rightarrow 出力は ABCXYZ となる。

12.3 反復不能編集記述子

(1) アポストロフィ、引用符および H 形編集記述子

これらは、ACOS FORTRAN 66 と同じ機能である。ただし、JIS FORTRAN 77 では入力に用いることと、引用符編集記述子は許されていない。

(2) T 形、TL 形および TR 形編集記述子

ACOS FORTRAN 66 では、記録の先頭から c 文字目の位置を Tc として指定できるのみであったが、TL 形および TR 形は、現在の位置から左へ c 文字目の位置を TLc 、右へ c 文字目の位置を TRc として指定する。ただし、 c はゼロでない符号なし整定数である。

出力の場合には、飛び越された部分で以前に文字が書かれていないところには、空白が出力される。入出力いずれの場合においても、 TLc によって記録中の最初の文字より前となったときには、最初の文字の位置を指定したものとみなされる。

例 次の 2 つの FORMAT 文は等価である。

```
FORMAT(T11, F5.0, 1X, F5.0, T1, I6)
FORMAT(TR10, F5.0, 1X, F5.0, TL21, I6)
```

(3) X 形編集記述子

TR 形編集記述子と等価である。すなわち、出力の場合には、飛び越された部分で以前に文字が書かれていないところには、空白が出力されるが、すでに文字が書かれているところは、変化しない。FORTRAN 66 では、必ず空白が出力されていた。

例 下記の WRITE 文の実行による出力は FORTRAN 77 と 66 で異なる。

WRITE(6, 1) } \Rightarrow FORTRAN 77 では ABCXYZ
1 FORMAT(1H_, 'ABC', T1, 4X, 'XYZ') } \Rightarrow FORTRAN 66 では _._._XYZ

(4) コロン編集記述子

入出力並びに項目が残っていない場合に、書式制御を終了させる。入出力並びに項目が残っている場合には、何の効果もない。

例 下記のWRITE文の実行による出力はコロン編集記述子の有無によって異なる。

WRITE(6, 1) I
 1 FORMAT(1H~, 2('I = ', I3, ':')) } \Rightarrow コロン有りのときは I = [I の値]
 ノコリの場合は I = [I の値] I =
 ↑
 有無

(5) S形、SP形およびSS形編集記述子

数値出力欄中の先頭の正の符号の出力を制御する。それぞれの書式付き出力文の実行開始時点において、数値出力欄中の先頭に正の符号を出力するか否かは処理系にまかされており、ACOS FORTRAN 77では正の符号は出力されない。しかし、書式仕様中のS P形編集記述子に出会うと、以後のI, F, E, D, Q およびG形編集記述子に対して、正あるいはゼロの数値の先頭にプラスが outputされる。SS形編集記述子に出会うと先頭のプラスは出力されない。S形編集記述子に出会うと処理系の標準にもどるが、ACOS FORTRAN 77ではS形とSS形は等価となる。

S, SP, SS形編集記述子は書式制御が終了するか、他のS, SP, SS形編集記述子に出会うまでも有効である。入力にこれらの編集記述子を書いてもよいが、何の効果もない。

(6) BN形およびBZ形編集記述子

数値入力欄中の先行する空白以外の空白の取扱いを指定する。それぞれの書式付き入力文の実行開始時点において、このような空白は、その装置に対して有効なBLANK指定子(11.5.2(9)参照)の値によって、ゼロとみなされるか無視される。しかし、書式仕様中のBN形編集記述子に出会うと、以後のI, F, E, D, QおよびG形編集記述子に対して、数値入力欄中のこのような空白は、BLANK指定子の値にかかわらず、すべて無視される。BZ形編集記述子に出会うと、このような空白はすべてゼロとみなされる。

空白を無視するとは、入力欄から空白を取り除き、残りの部分を右に寄せ、さらに先頭に空白を付け加えた形で解釈することと同じである。ただし、すべてが空白の欄はゼロとみなされる。

BN, BZ形編集記述子は書式制御が終了するか、他のBN, BZ形編集記述子に出会うまで有效である。出力にこれらの編集記述子を書いてもよいが何の効果もない。

(7) 斜線およびP形編集記述子

これらは、FORTRAN 66 では欄区切りおよび桁移動子と呼ばれていたものである。FORTRAN 77 では編集記述子として扱われるため、コンマで区切って書くことができる以外は、FORTRAN 66 と 77 の間に差異はない。

12.4 反復可能編集記述子

(1) I形編集記述子

I形編集記述子には Iw 形と $Iw.m$ 形の二種類があり、 Iw 形は FORTRAN 66 と同じ機能である。 $Iw.m$ 形は、入力では Iw 形と等価に扱われるが、出力では、符号なし整定数が少くとも m 個の数字から成り、必要ならば頭にゼロがつけられる。ただし、 m は符号なし整定数かつ $w \geq m$ でなければならない。 m がゼロかつ内部の値がゼロの場合は、S, SP および SS 形編集記述子にかかわらず、出力欄は空白となる。

例	内部の値	編集記述子	出力欄 (SP指定ナシ)	出力欄 (SP指定アリ)
1 2 3 4 5		I 5	1 2 3 4 5	* * * *
1 2 3 4 5		I 8.6	_ _ 0 1 2 3 4 5	_ + 0 1 2 3 4 5
0		I 5	_ _ _ _ 0	_ _ _ + 0
0		I 5.0	_ _ _ _ _	_ _ _ _ _

(2) F形編集記述子

ACOS FORTRAN 77 では、 $Fw.d$ において、 d が 36 を越えるときに d として 36 を指定したものとして取扱われる。ACOS FORTRAN 66 では、対応する入出力並びの型に依存していた。

例	入出力並びの型	書式中の編集記述子	実際に適用される編集記述子		
			ACOS	FORTRAN 77	ACOS
実数型または 倍精度実数型		F 30.20	F 30.20		F 30.18
		F 50.40	F 50.36		F 50.18
4 倍精度実数型		F 50.40	F 50.36		F 50.36

上記の d の取扱い以外は、ACOS FORTRAN 66 と 77 の間に差異はない。

(3) E形、D形およびQ形編集記述子

これらの編集記述子は、実数型、倍精度実数型、4 倍精度実数型、複素数型、倍精度複素数型および4 倍精度複素数型のいずれに対しても、使用することができる。 d の取扱いは、F形編集記述子と同じである。

FORTRAN 77 で追加された $Ew.dEe$ 形は、入力では $Ew.d$ 形と等価に扱われるが、出力では指数部を e 桁で表示する。ただし、 e はゼロでない符号なし整定数である。

例	内部の値	編集記述子	出力欄 (SP指定ナシ)
0.12345 × 10 ¹		E 12.5	_0.12345E+01
0.12345 × 10 ¹⁰⁰		E 12.5	_0.12345+100
0.12345 × 10 ¹⁰⁰		E 13.5 E 3	_0.12345E+100
0.12345 × 10 ¹⁰⁰		1PE13.5 E 3	_1.23450E+099

D形およびQ形編集記述子には指数部の表示桁を変更する機能はないが、 $Ew.dEe$ 形を倍精度お

より4倍精度に対して使用すればよい。

(4) L形編集記述子

入力において、.Tあるいは.Fで始まる文字列を与えてよいことと、すべてが空白の入力がエラーとなること以外は、ACOS FORTRAN 66と77の間に差異はない。

(5) A形およびR形編集記述子

A形編集記述子において欄の巾 w を省略することができる以外は、ACOS FORTRAN 66と同じ機能である。ただし、R形編集記述子は、JIS FORTRAN 77では許されていない。

A形編集記述子の欄の巾 w が省略されている場合には、欄の巾として、対応する入出力並びの項目の文字数が指定されたものとして取扱われる。

(6) G形編集記述子

G形編集記述子には Gw , $Gw.d$ および $Gw.dEe$ 形の三種類があり、 Gw 形と $Gw.d$ 形は ACOS FORTRAN 66と同じ機能である。ただし、JIS FORTRAN 77では Gw 形は許されておらず、 $Gw.d$ 形も、整数型、論理型および文字型には使用できない。

$Gw.dEe$ 形は、入力では $Gw.d$ 形と等価に扱われるが、出力では、 $Gw.d$ 形が $Ew.d$ 形と等価な範囲では $Ew.dEe$ 形と等価となり、 $Fw.d$ 形と4つの空白と等価な範囲では、 $Fw.d$ 形と($e+2$)個の空白と等価となる。ただし、 e はゼロでない符号なし整定数である。

12.5 並びによる書式(暗黙書式)

12.5.1 並びによる入力

並びによる入力における入力データの与え方は、ACOS FORTRAN 66と77では少し異っている。ACOS FORTRAN 77の解釈を以下に示す。

(1) 空白の取扱いと記録の終わり

BLANK指定子(11.5.2(9)参照)の値にかかわらず、空白はゼロとはみなされない。ACOS FORTRAN 66では空白がゼロとみなされる場合があった。

記録の終わりは、文字定数の中に現れない限り、空白と同じ効果をもたらす。

(2) 入力の値

入力の値は、定数、空の値、 $r*c$ または $r*$ の形とする。ただし、 r はゼロでない符号なし整定数、 c は定数とする。 $r*c$ は定数 c が r 個連続して現れたものとし、 $r*$ は空の値が r 個連続して現れたものと同じとする。ACOS FORTRAN 66では定数のみを与えることができた。空の値の意味については(8)を参照のこと。

(3) 値区切り子(区切り記号)

- ゼロ個以上の空白が前後に続いているコンマ。
- 二つの定数の間または定数に続く一つ以上の連続する空白。

- ゼロ個以上の空白が前後に続いている斜線。ACOS FORTTRAN66では不可。
ただし、次の場合は値区切り子の一部とはみなされない。
 - 文字定数の中の空白、コンマおよび斜線。
 - 複素定数の中の実部または虚部の前後の空白および、実部と虚部の間のコンマ。
 - 並びによる入力文によって読まれた最初の記録中の、直後にコンマや斜線が続かない先行する空白。

値区切り子の斜線は、直前の値を対応する入力並び項目に代入した後、その入力文の実行を終了する。入力並び項目が残っている場合は、それらには空の値（[\(8\)参照](#)）を与えたものとみなされる。

FORTRAN77ではコンマの前後に空白があっても一つの値区切り子であるが、ACOS FORTRAN66では二つの値区切り子として認識される場合があった。その例を次に示す。

例 READ(5, *) I, J } FORTRAN 77 では $I = 1, J = 2$ となる。
 1, 2 } ACOS FORTRAN 66 では $I = 1, J = 0$ となる。

(4) 実数型、倍精度実数型および4倍精度実数型の定数

指数を示すE、DおよびQの後の空白も許されていないこと以外は、ACOS FORTRAN 66と同じである。

(5) 複素数型、倍精度複素数型および4倍精度複素数型の定数

入力の形は、(実部, 虚部) である。ただし、実部と虚部の前後に空白があってもよい。実部と虚部は(4)で受入れられる形とする。記録の終わりは、実部とコンマの間またはコンマと虚部の間にのみ現れてもよい。

ACOS FORTTRAN 66では括弧で囲むことはせず、実部と虚部の各々が実数型、倍精度実数型、4倍精度実数型に対応していた。

(6) 論理型の定数

L形編集記述子（12.4(4)参照）で許される形とする。ただし、文字の中に空白、コンマおよび斜線を含んでいてはならない。

(7) 文字型の定数

入力の形は、' $h_1 h_2 \cdots h_n$ ' または" $h_1 h_2 \cdots h_n$ "である。ただし、 h_j は JIS コードで表現できる文字とする。アポストロフィ(引用符)で囲まれた h_j の中では一つのアポストロフィ(引用符)によって表現しなければならない。JIS FORTRAN 77 では引用符で囲む形は許されていない。

文字型の定数は、複数の記録にわたって書いててもよい。このとき、記録の終わりによって、空白またはその他の文字が定数の一部となることはない。

入力として与えた文字の数と、対応する入力並びの項目の文字数が一致しないときの処理は、

ACOS FORTRAN66と77の間に差はなく、A形編集記述子と同じである。

ACOS FORTRAN66と77では、アポストロフィまたは引用符で囲まなくてもよい場合がある。この場合の条件はかなり複雑であり、ACOS FORTRAN66と77の間でも差があるが、簡単に言えば、文字型の定数として一意的に解釈できるときである。すなわち、ACOS FORTRAN 77では、文字型の定数の中に、空白、コンマ、斜線、星印、アポストロフィおよび引用符を含まない場合と考えてもらってよい。

(8) 空 の 値

入力の形は次のいずれかとする。

- 間に文字をもたない連続する値区切り子。
- 並びによる入力文で読まれた最初の記録中の先行する文字をもたない最初の値区切り子。
- r^* の形。

空の値は、対応する入力並びの項目の定義状態に何の影響も与えない。すなわち、並びによる入力文の実行以前に確定であれば、その値は変化せず、不定であれば不定のままとなる。空の値を複素数関係の型に使用してもよいが、実部または虚部のいずれか一方には使用できない。

ACOS FORTRAN 66では空の値は存在しなかった。空白の後にコンマが現れた場合、あるいはコンマが連続して現れた場合は、対応する入力並び項目の型によって、ゼロ、偽または空白の値によって確定となっていた。

例 $I=1, J=2, K$ は不定のときに下記の入力文を実行するものとする。

READ(5, *) I, J, K	$\left\{ \begin{array}{l} \ldots, 1, , \\ \end{array} \right.$	FORTRAN 77 では $I=1, J=1, K$ は不定となる。 ACOS FORTRAN 66 では $I=0, J=1, K=0$ となる。
--------------------	----------------------------------------------------------------	--------------------------------------------------------------------------------

12.5.2 並びによる出力

並びによる出力により、出力される定数の形を次に示す。

出力並びの項目の型	出 力 さ れ る 定 数 の 形	
	ACOS FORTRAN 77	ACOS FORTRAN 66
整 数 型	I 15	I 16
実 数 型	0 PG 15.8 または 1 PE 15.7 注1)	0 PE 16.8
倍 精 度 実 数 型	0 PG 25.18 または 1 PD 25.17 注1)	0 PE 26.18
4 倍 精 度 実 数 型	0 PG 43.36 または 1 PQ 43.35 注1)	0 PQ 44.36
複 素 数 型	(実部, 虚部) 注2)	0 P 2 E 16.8
倍 精 度 複 素 数 型	(実部, 虚部) 注2)	0 P 2 D 26.18
4 倍 精 度 複 素 数 型	(実部, 虚部) 注2)	0 P 2 Q 44.36
論 理 型	L 1	L 2
文 字 型	項目の値がそのまま出力される。	同 左

注1) G形が実質上F形と等価な場合にのみ、G形で出力される。

注2) 実部と虚部の出力の形は、実数型、倍精度実数型および4倍精度実数型と同じ。

出力される記録は次の規則にしたがっている。

- 文字定数の前後を除いて、定数と定数の間には一つの空白が入る。ただし、JIS FORTRAN 77では、複数個の空白および、ゼロ個以上の空白が前後に続いているコンマが出力されてもよいことになっている。ACOS FORTRAN66では、定数が連續して出力されたが、前表に示すように、文字型以外の型では各々の定数は空白で始まっていた。
- 記録の先頭には、必ず行送りのために一つの空白が出力される。
- 複数の記録にわたって出力される場合の記録の終わりは、定数と定数の間、複素数関係の定数のコンマと虚部の間および文字型の定数の内部にのみ現れることがある。
- $r*c$ の形の出力はされない。ただし、JIS FORTRAN 77ではこの形の出力をしてもよいことになっている。

(以下次号)

FORTRAN 77 概説(1)の補足

大阪大学大型計算機センターニュース Vol. 11, No. 3 pp. 59 - 78(1981)

- p. 62 上から 6 行目に追加

アポストロフィ(引用符)で囲まれた h_j の中では、一つのアポストロフィ(引用符)は、間に空白を含まない二つの連続するアポストロフィ(引用符)によって表現しなければならない。

- p. 66 上から 19 行目から 22 行目の補足

本文に述べた仕様は、早くても昭和 57 年 5 ~ 6 月頃から使用可能であり、現在の仕様は JIS FORTRAN 77 と同一である。