

ALGOL 60 の入出力の一例*

(UNIVAC-1107/1108)

松 尾 昌 泰**

1. 序 論

UNIVAC-1107 ALGOL における入出力は Procedure によって行なわれている。おもなものは READ と WRITE であるが、CALL の際のパラメータの数とか順序が任意であるので、ALGOL 60 による普通の Procedure に較べて Flexibility をもっている。READ, WRITE は、Card, Magnetic tape, Drum, Printer に関するもので、他に POSITION, REWIND があって、Tape, Drum の Positioning とか Tape Rewind に関する Procedure である。

UNIVAC-1107 システムでは、原則として Tape, Drum などの高速度入出力装置は Intermediate file として用い、Card, Printer などの低速度入出力装置への入出力は、自動的に Drum を媒体として Off-line に行なうようになっている。したがって Tape とか Drum については、データの記憶方法は、Internal format であり、Card, Printer などは External format である。後者としては、Processor 固有の Free-format とプログラマが指定する Format がある。データの様式が特に問題でない場合は、Format を指定せず、Free-format を使うのが便利である。

Format の指定方法は、FORTRAN に似ており、使用も簡単である。これら Free-format の output と input, Formatted output と input, Magnetic tape と Drum についての output と input の順に解説を行なう。

2. Free-format の output

Free-format の output は Printer と Card punch に対して、次のように WRITE Procedure を、call することによって行なう。

```
WRITE (PRINTER, V(1), ..., V(N))
```

または、WRITE (V(1), ..., V(N))

* An Example of The Input-Output Procedures of ALGOL 60 (UNIVAC-1107/1108), by Masayasu Matsuo (Nippon Remington Univac Kaisha, Ltd.)

** 日本レミントンユニバック株式会社

ただし、V(I) は Expression または Array の Identifier である。これは印書の場合であるが Punch の場合も同様に行なう。

```
WRITE (PUNCH, V(1), ..., V(N))
```

actual parameter のとしてリストされた順に Expression の値を、12 スペースの Field に output する。Printer の場合は 1 行に 10 個の値を、Punch の場合には、1 枚に 6 個までの値を、次のような form で output する。

INTEGER Field に右づめに整数形で印書。

Leading 'O' は印書せず、負のときは Leading '-' を印書。

REAL -X. XXXX, -NN の形で印書。

これは -X. XXXX₁₀-NN を意味し、
- は符号を示し、負のときは -、正のときは、ブランク。

REAL 2 (精度が REAL の 2 倍のもの) output については、REAL と同じ。

BOOLEAN 'TRUE' または 'FALSE' を左づめ印書。

COMPLEX 実数部と虚数部を REAL の形で、二つの Field に印書。

他に英数字を扱う String があり、その値としては ' < ' を含まない Basic symbol の系列 > ' の形をしている。

3. Free-format の input

Free-format input は Card から、

```
READ (CARD, V(1), ..., V(N))
```

または READ (V(1), ..., V(N))

のように READ Procedure の call で行なう。Card 上には Source language form で、データを表示しておけばよい*。READ procedure の LIST すべてに対してデータを、連続した Card から読みこんでくる。次の READ の call では、前の Card のデー

* ALGOL 60 の 3104 は 1107 ALGOL では 3 & 4 であるが、入出力に限って 3, 4 とする。

タの残りは無視して次の Card を読む。また Card 全欄を scan する無駄をさけるために、データのあとに * をパンチしてよい。データのタイプが、Actual parameter のタイプと異なっていると、適当に変換して読みこむ。

4. LIST について

上述の READ とか WRITE に対する Parameter としての V(I) を List と呼ぶが、この List の要素として、Expression, List Identifier, Array Identifier, FOR List がある。ここでいう FOR List というのは、

```
FOR I=1 STEP 1 UNTIL 5 DO A(I)
```

のようなものを示し、1107 ALGOL では、これを

```
FOR I=(1, 1, 5) DO A(I)
```

とも書ける。たとえば、

```
REAL ARRAY A(1:10);
```

```
INTEGER I, J; ...
```

```
READ (FOR I=(1, 1, 5) DO A(I), J);
```

または、LIST Identifier を使用して、

```
REAL ARRAY A(1:10);
```

```
INTEGER J, J; ...
```

```
LIST L(FOR I=(1, 1, 5) DO A(I), J); ...
```

```
READ (L);
```

とも書くことができる。

5. Formatted output

プログラムで Form を指定したいときは、FORMAT を使用する。Format の Declaration は、

```
FORMAT F(X 10, D 7.2, A 1.1);
```

のような形をしており、この例では F が Format identifier で、カッコ内のものが、Format string であり、X 10 とか A 1.1 などとは Format phrase という。Format identifier は READ, WRITE Procedure の Actual parameter として使用される。

Phrase は 3 種類あり、その一つは印書する値をどのように編集するかを決める Editing phrase、印書とかスペースの挿入とか他の常数をその行に挿入したりする Non-editing phrase、第 3 番目に繰返しのための Repeat phrase がある。まず Non-editing phrase には、次の四つがある。

Aw.d ... この Phrase のある時点で w 行紙をすすめて印書し、再び d 行すすめる。

Ew ... 第 $(w-1)$ 行目に Page eject する。

Xw ... 次の w 行の Skip (Expunge)。

'ABC' ... String 常数自身を挿入する。

Editing Phrase には次の六つがある。

Bw TRUE または FALSE を w の Field に左づめで印書。

Dw.d 小数点を入れて数値を Field w に右づめで印書。小数点以下 d 桁。

Iw.d 整数形 (d 進法) で Field w に右づめで印書。

Rw.d $-X.XXXXXX$, $-NN$ (これは $-X.XXXXXX_{10}-NN$ を示す) の形式で印書する。 d は有効桁数を示す。

Sw String の最初の w characters を左どめで印書。

Tw.d (Truncated) 有効 d 桁を Field w に右づめで印書。

次に Repeat Phrase について例で説明する。まず一つの Phrase R 12.9 を 3 回繰返すには、3 R 12.9 とし、Phrase のグループ R 12.9, X 5 を 4 回繰返えすには、4(R 12.9, X 5) とする。繰返えし回数が変数 N のときは、 $:N: (R 12.9)$ とし、 $N < 1$ のときは、カッコ内の Phrase は無視される。最後に Repeat 回数が不定のときは、カッコで囲んで、たとえば

```
FORMAT FM (X 4, (D 7.2, A 1.1), I 4)
```

とすると、I 4 の次に印書すべきものがなくなるまで、D 7.2 と A 1.1 を繰返えす。

6. Formatted input

READ procedure についても、WRITE と同様に Parameter として FORMAT をもってよい。Card 上の情報は Field の長さを指定することで、情報の delimiter は必要としない。FORMAT declaration は output の場合と同様で、Editing phrase, Non-editing phrase についても output の場合と大差ない。Editing phrase には他に F_w があり、Card 上に Free-format でパンチされた一つの情報を読むときに用いるものである。Format には関係ないが、Card input における End of File 条件について説明する。

ALGOL プログラムで読み込みうるものに、Data card 以外の Control card (特別なパンチがしてある) があり、また EOF (End of File) Card と EOF 以外の Control card の 2 種類がある。この Card により、プログラム処理を変更することができる。た

例えば

```
BEGIN LOCAL LABEL EN, FIN; ...
READ (CARDS, DATA, EN, FIN); ...
FIN: WRITE ('OWARI'); ...
EN: WRITE ('EOF ON READER'); ...
END;
```

とすると、EOF Card があれば Parameter の最初の Label (EN) に、他の Control card があれば、2 番目の Label (FIN) に Control が移される。

8. Magnetic tape と Drum の入出力

READ, WRITE procedure の call は次のようにして行なう。

```
WRITE (d, m, l, f, o);
```

```
READ (d, l, f, i);
```

ただし、d は Device を示し、PRINTER, CARD, PUNCN についてはすでに述べた。o, i, f はそれぞれ Output-List, Input-List, Format を示すものでこれについても説明した。

Device としては TAPE (I) と書くが、この I が 0~5 は Tape を、6, 7 のときは Drum を、8 のときは Printer を、9 のときは Punch を示す。

TAPE (8), TAPE (9) は PRINTER, PUNCH と同じ意味をもつ。この Procedure を 1 回 call することにより、1 Record を読み書きするので、READ Procedure の Input-list は、普通その Record を、output した Output-list と対応させてつかう。

次に Parameter に l とあるのは Label-list のことで、Procedure での条件によって戻される Label を指定する。たとえば、WRITE のときには、二つの Label を用い、一つは Tape の Physical end のときに、他一つは Internal block limit に達したときに、戻される Label である。

Tape とか Drum に二つ以上の File を入れるとき、一つの File 内でのデータの区別を行なうときには、Modifier によってその区別を行なう。Modifier として、EOI (End of information), EOF, EOF (e), KEY, KEY (e) がある (e は Expression)。このうち EOI, EOF, EOF (e), は READ Procedure で判別でき、KEY, KEY (e) は Tape (drum) の Positioning のときに用いるものである。Positioning は、

```
POSITION (TAPE(I), p, l)
```

で行なう。ただし、p は Positioning parameter を、すなわち Modifier を示し、その Modifier がある所に Positioning される (逆方向の Positioning 可能)。

Tape (Drum) の Rewind Procedure は

```
REWIND (TAPE(I), TAPE(J), ...);
```

で call する。Interlock つきで rewind するには次のようにする。

```
REWIND (TAPE(I), ..., INTERLOCK);
```

これまでに述べた READ, PUNCH, CARDS, TAPE, DRUM, EOF, KEY, EOI, WRITE, POSITION, REWIND, PRINTER はすべて Library procedure であり、Declare なしで使用できるものである。

9. 例

最後に $N \times M$ の行列 A の要素を印書することを考える。各要素の印書様式は、R 12.6 とすると、次のようなコードになる。

```
REAL ARRAY A (1: N, 1: M);
INTEGER II, JJ, K;
LIST MAT (EOR II=(1, 10, N) DO
  (FOR JJ=(II, 1, MIN(II+9, N))
  DO JJ, FOR K=(1, 1, M) DO
    (K, FOR JJ=(II, 1, MIN(II+9, N))
    DO A (JJ, K)))));
FORMAT KATA (XII, : ENTIER (MIN
  (II+9, N))-II+1: ('COLN', I4, X4), A1,
  :M: ('ROW', I3, : ENTIER (MIN(II+9,
  N))*-II+1: (R 12.6), A1), A2);
WRITE(KATA, MAT); ...
```

参考文献

- 1) Peter Naur (Editor): Revised Report on the Algorithmic Language, CACM, 6, 1 (1963), pp. 1~17
- 2) UNIVAC 1107 Software Reference Manual [6], ALGOL, pp. 32~45
- 3) D.E. Knuth (Chairman): A Proposal for Input-Output Conventions in ALGOL 60, CACM 7, 5 (1964), pp. 273~283
- 4) H. Hoffmann: A Proposal for Input-Output Handling in ALGOL 60, IFIP ALGOL Bulletin (1965)

(昭和 40 年 4 月 29 日受付)

* ENTIER (MIN (II+9, N)) とは II+9 と N の小さい方をとり出して (実数型) になるそれを整数型にすることである。