日本語SNOBOL4の使用について

吉田, 和幸
九州大学工学部情報工学科

牛島, 和夫
九州大学工学部情報工学科

https://doi.org/10.15017/1468102

出版情報：九州大学大型計算機センター広報. 17 (3), pp.110-134, 1984-05-25. 九州大学大型計算機センター
バージョン：
権利関係：

研 究 開 発

日本語ＳＮＯＢＯＬ４の使用について

吉田　和幸＊，牛島　和夫＊＊

<< 目 次 >>

昭和５９年３月２９日受理

＊九州大学工学部情報工学科（現在，大分大学工学部組織工学科）
＊＊九州大学工学部情報工学科

目次中※印がある項目は前回の日本語ＳＮＯＢＯＬ４紹介記事（広報 Vol.16, №2［５］）の以後に追加・変更があった項目である。

この文書はＪＥＦによる原稿を写真版にしたものである。

1．はじめに

最近では，計算機により日本語文字（漢字，ひらがな，カタカナ，ローマ字および日本語の文章に使用する特殊文字をまとめてこう呼ぶ）を扱うことが当然のようになってきた．最近国内で発表される計算機はそのほとんどが「漢字が扱えます」とうたっている．しかし，実際に日本語文字を扱うプログラムを書こうとすると，計算機の細部まで知っている必要があったり，面倒な手順を要求されたりする場合が多い．

九州大学大型計算機センターでは１９８０年から富士通の提供するＪＥＦ（Ｊａｐａｎｅｓｅ　Ｐｒｏｃｅｓｓｉｎｇ　Ｅｘｔｅｎｄｅｄ　Ｆｅａｔｕｒｅ）［1］と言う日本語情報処理機能を導入している．ＪＥＦの利用形態として最もよく行われているものの一つに，日本語ワードプロセッサとしての利用がある．しかしもっと高度な使い方（たとえば［2］にある文書作成支援ツールのようなこと）をしたければ，自分でプログラムを組む必要が出てくる．このためにはＣＯＢＯＬ，ＦＯＲＴＲＡＮ，ＰＬ／Ｉの三つの言語でＪＥＦの機能を使えるようになっている．ところが，これらの言語は非常に「手続き的」であったり，固定したフォーマットのファイルしか扱えなかったりするので，日本語文字処理を伴うプログラムをこれらの言語で書くのは，かなりむずかしい．しかもＥＢＣＤＩＣ文字（１バイトコード）を処理するための従来からある文字型とは別に日本語文字（２バイトコード）を処理する日本語文字型を導入しているため，これらの２種のコード系が混在するＦＤＭＳ日本語文章ファイルの処理は一層困難である．

ＳＮＯＢＯＬ４［3］は挿入，削除を伴ったパターンマッチを一文で書け，その組み合わせにより文字列に関するかなり複雑な操作も簡単に記述できる．従ってテキストファイル処理などに際して適当なツールがない場合その場でプログラムを作ってみるというラピッドプロトタイピングに適した言語である［4］．

我々はこのＳＮＯＢＯＬ４に日本語テキスト処理機能を追加した日本語ＳＮＯＢＯＬ４を作成し，１９８３年３月九大大型計算機センターで公開した［5］．この後，文字列パターンマッチ処理の効率改善，日本語文章ファイルへの出力機能の追加等を行った［6，7］．ここではあらためて日本語ＳＮＯＢＯＬ４全般について説明する．

日本語ＳＮＯＢＯＬ４（Ｓ４Ｊと略記する）は，日本語文字の処理を可能とするばかりでなく，ＴＳＳで利用しやすくするための改良も行った．既存のＳＮＯＢＯＬ４に対して追加した機能は次の通り（番号の前に▷印があるものは今回新たに加わった機能である）．

① 可変長ファイルの入出力ができる．

▷② 日本語文章ファイル（ＦＤＭＳ和文エディタ［8］で作成したもの）の入出力ができる．

▷③ 変数名，関数名，名札，リテラルに日本語文字が使用できる．日本語文字とは漢字，ひらがな，カ

タカナ，ギリシャ文字，ロシア文字，ローマ字，数字からなる2バイトのコードで表す文字を指す．

▷④ 日本語文章ファイルの処理に便利なキーワードを追加した．

⑤ コマンド行中にプログラムへ渡す文字列パラメータを書ける．

▷⑥ 端末からデータを入力する際に出力するプロンプタを自由に設定できる．

⑦ プログラムの実行時にプログラム中からＴＳＳコマンドを実行できる．

⑧ ＦＯＲＴＲＡＮ７７で書いた外部関数との間で文字列型のデータの受け渡しもできる．

既存のＳＮＯＢＯＬ４から変更になった仕様は次の３点である．

⑨ ソースプログラムは行番号の付いた可変長ファイルあるいは行番号なしの固定長ファイルであること．どちらの場合もプログラムは1行に７２文字以内に収めること．

⑩ 装置機番ｕｕに対するＤＤ名は，ＳＮＯＢ００ｕｕである．

⑪ ＯＵＴＰＵＴ関数のフォーマット指定はできない．

①②⑩⑪については「２．入出力の方法」で，④については「３．日本語テキスト処理用のキーワード」で述べる．③⑨については「４．プログラムの書き方」を，⑤⑥⑦は「９．ＴＳＳ環境との通信」を，⑧は「１０．外部関数」を参考にしてほしい．

ＳＮＯＢＯＬ４はわが国ではあまり知られていないが，米国では比較的しろうと向けの言語として人文科学の分野を中心に使われているようだ．ＳＮＯＢＯＬ４プログラミングを学ぶには文献［３］がある．日本語で書かれた入門向けの教科書は，ほとんど見当たらない．以前広報にＳＮＯＢＯＬ４の紹介を書いたのでそれが案内になるかもしれない［９］．

ＳＮＯＢＯＬ４によるプログラミング集が一冊の本になっている［１０］．これは文字列操作の各種アルゴリズム，文字列と配列との相互変換，リスト処理，パタン照合，簡単な文書処理，各種整列法，いくつかのゲームなどのアルゴリズムをＳＮＯＢＯＬ４で書いて一冊にまとめたものだ．自分の問題をＳＮＯＢＯＬ４で解決しようという人に参考になるだろう．ここにあげられたプログラムの大部分はこの説明書で紹介する日本語ＳＮＯＢＯＬ４処理系の上でそのまま用いることができるので重宝である．

２．入出力の方法

既存のＳＮＯＢＯＬ４の入出力は，固定長ファイルのみを対象としていたが，Ｓ４Ｊでは可変長ファイル，日本語文章ファイルも扱えるようになった．これにともない，入出力が以下のように変更になった．

２．１ ＤＤ名

機番ｕｕに対応するＤＤ名を変更した．下の例のように，従来は'ＦＴｕｕＦ００１'であったが，Ｓ４Ｊ

では'SNOB00uu'となる.

表2－1. 装置機番に対応するDD名の例

機番	S4JのDD名	従来のDD名
3	SNOB0003	FT03F001
14	SNOB0014	FT14F001

機番の範囲は1≦uu≦39あるいは50≦uu≦99である. 40～49の機番はS4J処理系が使用する. 30～39の機番は日本語文章ファイル専用としている. 一般ファイル用の機番としてこの他の1～29と50～99が使用できる.

2. 2 OUTPUT関数

従来, OUTPUT関数では, 第3引数に'(1H ,136A1)'のようなFORTRANのフォーマット記述子を書いていたが, S4Jではこれを廃止した. かわりに各行の先頭にラインプリンタ用の制御文字を1文字付加できるようにした. 付加すべき文字はOUTPUT関数の第3引数で指定する.

表2－2. 機番20を変数OUTと結合する場合のOUTPUT関数

付加したい文字	S4JのOUTPUT関数
空白（行送り）	OUTPUT(.OUT,20,' ')
1（改ページ）	OUTPUT(.OUT,20,'1')
付加しない	OUTPUT(.OUT,20)

2. 3 可変長ファイルの取り扱い

可変長ファイルの入出力は, 次の点を除き固定長ファイルと同様に行える.

INPUT関数の第3引数は, 1行の最大文字数（バイト数ではない）と解釈する. 入力時にここに指定したより多くの文字が1行に存在すると, 余りは切り捨てる. 1行の文字数が指定より少ないときは, 実際に読み込んだ文字列を入力変数の値とする. 空白等の詰め込みは行わない.

2. 4 日本語文字の入ったファイルの入出力

・右筆で扱える形式のファイルへの入出力および日本語ラインプリンタへの出力は, 通常のファイルを扱う

方法と同様に行える．ただし，日本語文字が入ると１行の文字数とバイト数が異なることに注意すること．ＴＳＳのALLOCATEコマンドやATTRIBUTEコマンドではバイト数を指定する．

・日本語文章ファイルの入力には，機番３０～３９を用いる．ＩＮＰＵＴ関数の第３引数には２５６以上の数を指定する．

```
INPUT(.NFILE,31,300)
```

この後，入力変数（ここではNFILE）を引用すると，日本語文章ファイルから読み込んだ文字列が次の形式で得られる．

```
PPUUUIIIddddddd・・・
```

ここでPP，UUU，IIIはＦＤＭＳ和文エディタで画面の左に出る「参照番号」と呼ばれる数字である．ｄｄｄｄｄｄ・・・が実際のデータで９文字目から始まる．日本語文章ファイルからの入力の際には実際に読み込んだ文字数が入力変数の長さになり，空白のつめ込みは行わないので注意してほしい．

例：

```
01008000２．@UL@入出力の方法@UE@@NL@
01009000　既存のＳＮＯＢＯＬ４の入出力は，固定長ファイルのみを
```

従って，データ部だけを取り出すには次のパターンマッチを実行すればよい．

```
NFILE LEN(8) . REFNO REM . NBUF  :F(EOF)
```

REFNOには参照番号が，NBUFには日本語文章データが入る．

日本語文章ファイルへの出力は，その入力の場合と同様に機番３０～３９を用いる．この場合，ＯＵＴＰＵＴ関数の第３引数は指定しないこと．

```
OUTPUT(.NWFILE,32)
```

この後に出力変数（ここではNWFILE）にデータを代入すると（一般のファイルへの出力と同様に）日本語文章ファイルへの出力を行う．参照番号等の制御情報は，ＳＮＯＢＯＬ４処理系が生成し，データに付加する．日本語文章ファイルへ出力するデータ中にEBCDIC文字が出現するときは，必ず@で囲まなければならない．なぜならば，ファイル中のデータを編集するために和文エディタを使用する時に，和文エディタが@で囲まれていないEBCDIC文字列を検出し，それらを消去してしまうためである．

３．Ｓ４Ｊで追加・変更したキーワード

Ｓ４Ｊでは日本語テキスト処理用に９個のキーワードを追加した．これにともない，従来からあるキーワード（&ALPHABET）の仕様を変更した．

３．１　日本語文字を字種別に分類するためのキーワード

日本語テキストを処理する際に文字種の情報は役立つ．Ｓ４Ｊでは字種ごとに表３－１に示すキーワードを用意している．&ALPHABETは従来はＳＮＯＢＯＬ４で使用できるすべての文字をそのコード順に並べたものを意味していたが，Ｓ４ＪではEBCDIC文字だけをコード順に並べたものを意味する．字種分類用のキーワードはSPAN関数，BREAK関数等の中で使用する．たとえばこれらのキーワードを使用して漢字列を取り出すプログラムは次のようになる．

```
      LINE = '日本語テキスト処理用のキーワード'
LOOP  LINE SPAN(&KANJI1 &KANJI2) . WORD =    :F(END)
      OUTPUT = WORD                          :(LOOP)
END
```

このプログラムを実行すると

日本語
処理用

という出力が得られる．なお，各キーワード中では文字はそのコードの昇順に並んでいる．

表３－１．日本語文字を字種別に分類するためのキーワード

キーワード	内　容
&KANJI1	ＪＩＳ第１水準漢字（２９６５字）
&KANJI2	ＪＩＳ第２水準漢字（３３８４字）
&HIRAGANA	ひらがな（８３字）
&KATAKANA	カタカナ（８６字）
&GREEK	ギリシャ文字（大文字，小文字合わせて４８字）
&RUSSIAN	ロシア文字（大文字，小文字合わせて６６字）
&ALPHABET	ＥＢＣＤＩＣ文字（２５６字）

３．２　EBCDIC文字に対応する日本語文字を対応する順に並べたもの

日本語文字のうちローマ字，数字および一部の特殊記号はEBCDIC文字に対応するものがある．これら

の文字とEBCDIC文字との相互変換の機能があると日本語文章ファイルを扱う上で便利である．このためにＳ４Ｊでは表３－２に示すキーワードを用意している．

表３－２．日本語文字とEBCDIC文字との相互変換のためのキーワード

キーワード	内 容
&NALPHABET	日本語文字のローマ字，数字，特殊記号を対応するEBCDICコードの順に並べたもの（２５６字）

この&NALPHABETは&ALPHABETと組み合わせてREPLACE関数とともに用いる．たとえば変数LINE中のEBCDIC文字列を日本語文字列に変換するときは次のように書く．

```
LINE = REPLACE(LINE, &ALPHABET, &NALPHABET)
```

EBCDIC文字から日本語文字へ変換するときは上の例でREPLACE関数中の&ALPHABETと&NALPHABETを入れ換える．

３．３ 日本語文字の大きさを設定するキーワード

日本語ラインプリンタ（NLP）は日本語文字を出力する際にその大きさを９ポイントと１２ポイントの２種類選べる．ＳＮＯＢＯＬ４プログラムからNLPへの出力の際にこの２種類の大きさの文字を選べるようにＳ４Ｊでは表３－３に示すキーワードを用意している．

表３－３．日本語文字の大きさを設定するキーワード

キーワード	内 容
&NMODE	日本語文字を出力する際の大きさを設定する．その値は９または１２．（初期値は１２）

たとえばこのキーワードを用いて文字の大きさを変更するプログラムは次のようになる．

```
      LINE = '日本語TEXT処理用のKEYWORD'
      OUTPUT = LINE
      &NMODE = 9
      OUTPUT = LINE
END
```

このプログラムを実行すると

日本語TEXT処理用のKEYWORD
日本語TEXT処理用のKEYWORD

という出力が得られる.

3．4　S４Jコマンドからのパラメータを受け取るためのキーワード

S４Jコマンド（S４Jを起動するためのTSSコマンド，5．参照）からのパラメータを受け取るために表3－4に示すキーワードを用意している.

表3－4．S４Jコマンドからのパラメータを受け取るためのキーワード

キーワード	内　容
&PARM	S４Jコマンドに書いたパラメータが設定される

&PARMについては9．1で詳説する.

4．プログラムの書き方

S４Jのソースプログラムは，行番号付き可変長データセットとして作成する必要がある．可変長データセットを作成するもっとも簡単な方法はTEXTファイルとしてEDITコマンドで作成する方法である.

```
例：　E ABC.TEXT NEW          ← 順編成データセットの場合
      E ABC.TEXT(XYZ) NEW     ← 区分編成データセットの場合
```

ソースプログラムは行番号部分（先頭の8文字．ただしEDITコマンドではそのうち5文字しか表示しない）を除いて72文字以内に収まっていなければならない．72文字を超えた部分は無視されてしまう.

S４Jでは，変数名，関数名，名札，（総称して名前と呼ぶ）に，従来のEBCDIC文字のほかに，日本語文字中の漢字，ひらがな，カタカナ，数字，ローマ字，ギリシャ文字，ロシア文字を用いることもできる．名前（identifier）の構文は図4－1に示すものになる．名前に日本語文字を使用した場合，EBCDIC文字で書いたおなじ綴りの名前とは区別されるので注意してほしい．とくにキーワードやシステムが定義している名前はすべてEBCDIC文字であること．日本語文字で

```
OUTPUT = 'これは日本語文字'
```

と書いても単に変数OUTPUTに代入が起こるだけで，出力は行われない.

文字リテラル中にはすべての日本語文字を書くことができる.

日本語文字の＾？¥．！％＊／＃＋－@｜&はEBCDIC文字の演算子（¬?¥.!%*/#+-@|&）の代

わりに使用できる．この場合，日本語文字の演算子とEBCDIC文字のそれとは同等に扱い，区別しない．

```
digit   ::=   0|1|2| . . . |9   (EBCDIC文字)
            |０|１|２| . . . |９   (日本語文字)
letter  ::=   A|B|C|. . . |Z   (EBCDIC文字)
        |Ａ|Ｂ|Ｃ| . . . |Ｚ|ａ|ｂ|ｃ|. . . |ｚ        (日本語文字)
        |ぁ|あ|ぃ|い|ぅ|う| . . . |ん
        |ァ|ア|ィ|イ|ゥ|ウ| . . . |ン|ヴ|ヵ|ヶ
        |Α|Β|Γ| . . . |Ω|α|β|γ| . . . |ω        (ギリシャ文字)
        |А|Б|В| . . . |Я|а|б|в| . . . |я        (ロシア文字)
        |亜|唖|娃| . . . |腕     (ＪＩＳ第１水準漢字)
        |弌|丐|丕| . . . |鷲     (ＪＩＳ第２水準漢字)
alphanumeric ::= letter | digit
identifier ::= letter {alphanumeric | . | _ }
```

図４－１．名前（identifier）の構文

5．日本語ＳＮＯＢＯＬ４の使用法

ここでは日本語ＳＮＯＢＯＬ４（Ｓ４Ｊ）のＴＳＳでの代表的な使用法について述べる．詳しいことは「6．ＴＳＳコマンドおよびカタログドプロシジャ」で述べる．Ｓ４Ｊコマンドの最も基本的な使用法は，

```
S4J  データセット名
```

である．たとえば，

```
S4J ABC.TEXT
```

とすると，データセットABC.TEXT中のＳＮＯＢＯＬ４プログラムを実行する．この時，Ｓ４Ｊ処理系は２種類の出力を生成する．１つは図５－１に示す画面への出力であり，これには実行時に検出したエラーと統計情報が書いてある．他の１つは図５－２に示すプロファイルリストと呼ばれるものである．これはソースプログラムにその各文の実行回数，成功回数，失敗回数を付加したものである．ソースプログラム中に文法的なエラーがあった場合，そのメッセージもここに出力する．とくに指定しないかぎりS4J.TEMP.LISTという名のデータセットへ出力されるので，実行後LISTコマンドで見ることができる．Ｓ４Ｊコマンドでも最後にそのＬＩＳＴコマンドを実行して画面にプロファイルリストを出力する．

```
NORMAL TERMINATION AT LEVEL    0
LAST STATEMENT EXECUTED WAS     1700
SNOBOL4 STATISTICS SUMMARY-
             20 MS. COMPILATION TIME
           1472 MS. EXECUTION TIME
           5176 STATEMENTS EXECUTED,     170 FAILED
           2331 ARITHMETIC OPERATIONS PERFORMED
           2112 PATTERN MATCHES PERFORMED
              7 REGENERATIONS OF DYNAMIC STORAGE
            169 READS PERFORMED
            557 WRITES PERFORMED
           0.28 MS. AVERAGE PER STATEMENT EXECUTED
```

図5－1. ＳＮＪ処理系からの出力（その１　統計情報）

```
   FACOM OSIV/F4 SNOBOL4(BELL TELEPHONE LABORATORIES) VER.3.11 + PROFILE + JEF (-830707-)     DATE 84/03/29    TIME 10:19:25

EXECUTION =  SUCCESS +  FAILURE :          S O U R C E   P R O G R A M

                                00000100*-------------------------------------*
                                00000200*     例題 : 漢字列計数               *
                                00000300*           1983年11月4日  改良       *
                                00000400*-------------------------------------*
         1           1           0 00000500           INPUT(.JINPUT,31,300)
         1           1           0 00000600           OUTPUT(.JOUTPUT,4)
         1           1           0 00000700           回数 = TABLE(30,10)
         1           1           0 00000800           漢字 = &KANJI1 &KANJI2
       169         168           1 00000900読み込み   テキスト = JINPUT                          :F(印刷)
       168         168           0 00001000           テキスト LEN(8) REM . テキスト            :F(読み込み)
      1944        1776         168 00001100単語探索   テキスト ARB SPAN(漢字) . 単語 =           :F(読み込み)
      1776        1776           0 00001200           回数<単語> = 回数<単語> + 1               :(単語探索)
         1           1           0 00001300印刷       JOUTPUT = '＊＊＊＊＊  漢字列計数結果  ＊＊＊＊'
         1           1           0 00001400           JOUTPUT =
         1           1           0 00001500           回数 = CONVERT(回数,'ARRAY')              :F(終了)
         1           1           0 00001600           I = 1
       556         555           1 00001700単語印刷   JOUTPUT = 回数<I,1> ' = ' 回数<I,2>      :F(終了)
       555         555           0 00001800           I = I + 1                                 :(単語印刷)
         1           1           0 00001900終了
         1           1           0 00002000END
NO ERRORS DETECTED IN SOURCE PROGRAM
```

図５－２．　Ｓ４Ｊ処理系の出力（その２　実行回数情報）

プロファイルリストが必要でない場合には，NP (NOPROFILE) オペランドを指定することにより，プロファイル機能を止めることができる．

例：　S4J ABC.TEXT NP

こうするとプロファイルリストはつくらず，ソースプログラムと文法エラーのメッセージを直接画面に出力する．

ソースプログラムのリストも必要でない場合は，さらにNL (NOLIST) を指定する．

例：　S4J ABC.TEXT NP NL

統計情報も出力したくないという場合にはNS (NOSUMMARY) を指定する．

例：　S4J ABC.TEXT NP NL NS

しかしながら，これらの情報は思いがけないエラーのデバッグに役立つことがあるので，よほどのことがない限り出力するようにしておく方がよい．

大量のデータを扱うプログラムでは，使用するメモリ領域の大きさを増やす必要がある．これは次のようにする．

S4J データセット名 L(最小メモリ量) H(最大メモリ量)

少なくとも「最小メモリ量」（単位はキロバイト）の領域を使い，可能ならば「最大メモリ量」（単位はキロバイト）を上限として，確保できる限り大きな領域を使うことを意味する．たとえば，

S4J ABC.TEXT L(100) H(2000)

とすると，少なくとも１００ＫＢのメモリの使用が保障され，可能であれば，２０００ＫＢまでのメモリが使用可能となる．なお，ＴＳＳで使用できるメモリ量は指定を省略すると，２０４８ＫＢ（ＬＯＧＯＮ時にＳＩＺＥオペランドで指定すれば５１２０ＫＢまで可能）となっているため，Lオペランドでこれを超える量を指定すると，異常終了する．LやHを指定しない場合には，「最小メモリ量」＝３０ＫＢ，「最大メモリ量」＝２００ＫＢとなる．

通常，変数INPUTは端末からの入力変数に割り当てられている．Ｓ４ＪコマンドのＩＮＰＵＴオペランドによりこれをデータセットからの入力に切り替えることができる．指定法は，

S4J ソースプログラムデータセット INPUT(入力データセット)

ただし，変数INPUTは初期状態では入力文字数を８０に設定している．１行に８０文字以上データがある場合には注意すること．

多くのファイル（約３０以上）を同時に使用したためにバッファ領域が不足する場合，あるいは外部関数

用のメモリ領域が足りないときにはRオペランドを指定する．ユーザが作成した外部関数を使用する場合にはＬＩＢオペランドを使う．ＰＡＲＭオペランドを指定すると任意の文字列をプログラムに渡せる．

プログラムの実行を途中で打ち切るためにはアテンションキー（ブレイクキー，ＰＡ１という端末もある）を用いる．プログラムの実行中に１回アテンションキーを押すとＳ４Ｊ処理系は，

CUT BY SYSTEM IN STATEMENT ・・・

というメッセージを出力してプログラムの実行を打ち切る．そのさいにそこまでの統計情報と実行回数情報（プロファイルリスト）とを出力する．統計情報の出力が終了し，プロファイルリストが画面に出始めるまでは，再度のアテンションキーの打鍵はしないこと．プロファイルリストの画面への出力は，アテンションキーで止められる．

6．ＴＳＳコマンドおよびカタログドプロシジャ

6．1 Ｓ４Ｊコマンド

Ｓ４Ｊコマンドは，Ｓ４Ｊで書いたプログラムをＴＳＳ環境で動かすためのコマンドである．Ｓ４Ｊコマンドの構文を以下に示す．

```
S4J ソースプログラム [LIST(プロファイル)]  [INPUT(実行時データ)]
                     [NOLIST             ]

    [LIBRARY(ライブラリ)]  [PARM('文字列')]

    [LOW(最小メモリ数)]  [HIGH(最大メモリ数)]

    [RESERVE(予約メモリ数)]

    [PROFILE  ]  [SUMMARY  ]
    [NOPROFILE]  [NOSUMMARY]
```

［α］　は，項目αが省略可能であることを示す．

$\left[\begin{matrix}\alpha \\ \beta\end{matrix}\right]$ は，項目α又はβのどちらか一方を指定するか，全て省略するべきであることを示す．

αβγδと下線が引いてあるのは，αβγδと書くところをαγと略記できることを示す．

・「ソースプログラム」は，Ｓ４Ｊソースプログラムの入ったデータセット名を指定する．

・「プロファイル」は，プロファイルリストを出力すべきデータセット名を指定する．プロファイルリスト不要の場合はＮＯＬＩＳＴを指定する．

・「実行時データ」は，機番５からの入力となるべきデータセット名を指定する．

・「ライブラリ」は，外部関数ライブラリの入ったデータセット名を書く．

・「文字列」は，プログラムへ渡すべき文字列を指定する．プログラム中でキーワード&PARMを参照すると，この文字列を受け取ることができる（9．1参照）．

・「最小メモリ数」は，実行時に最低必要となるメモリの量をＫＢ（キロバイト）単位の整数で指定する．

・「最大メモリ数」は，可能なら確保したいメモリの量を指定する．

・「予約メモリ数」は，実行回数計数機能（プロファイル機能），入出力バッファ，および外部関数が使用するメモリの量である．（通常の使用では指定の必要はないであろう．）

・実行回数計数機能を働かせたくない時にはＮＯＰＲＯＦＩＬＥを指定する．

・実行終了後に出力される統計情報を出力したくない時にはＮＯＳＵＭＭＡＲＹを指定する．

各オペランドの省略値は次のように設定している．

プロファイル	: S4J.TEMP.LIST
実行時データ	: 端末からの入力
ライブラリ	: 'LIB.S4EXTLIB'
文字列	: 空文字列
最小メモリ数	: 35(KB)
最大メモリ数	: 200(KB)
予約メモリ数	: 100(KB)
他のオプション	: PROFILE,SUMMARY

データセット指定時の注意

Ｓ４Ｊコマンドはコマンドプロシジャとして作られている．このため，「プロファイル」，「実行時データ」，「ライブラリ」に他の課題番号のデータセットを指定する場合は，下の例のように３重の引用符で囲む必要がある．

```
S4J ABC.TEXT LIB('''SYS1.FORTLIB''')
```

また，「実行時データ」と「ライブラリ」には、複数のデータセットを指定できる．（ただし，各データセットのブロック長が同じであること．）

この場合，下のように指定する．

自分のデータセットを連結する時の書き方（LIBオペランド）：

```
LIB('MYLIB.LOAD MYLIB2.LOAD')
```

自分のデータセットと他の課題番号のデータセットを連結する時：

```
LIB('MYLIB.LOAD ''SYS1.FORTLIB''')
```

課題番号が自分と違うデータセットどうしの連結：

```
LIB('''SYS1.FORTLIB'' ''SYS1.TACLIB''')
```

6．2 カタログドプロシジャ（S４J）

S４Jをバッチ環境で動かすためのカタログドプロシジャS４Jを以下に示す．

```
プロシジャ名          パラメータ
     S4J              [LIB='ライブラリ',]
                      [PARA='文字列',]
                      [OPTIONS='[LIST   ]
                               [NOLIST ],
                               [PROFILE  ]
                               [NOPROFILE],
                               [SUMMARY  ]
                               [NOSUMMARY],
                               [HIGH=最大メモリ数,]
                               [LOW=最小メモリ数,]
                               [RESERVE=予約メモリ数] ']
```

・関連するDD名

SYSIN　　S４Jソースプログラムの入ったデータセットを指定する．

INPUT　　実行時データの入ったデータセットを指定する．

・各オペランドの意味および省略値はS４Jコマンドの対応するオペランドと同じである．ただしプロファイルリストは常にLPに出力される．ライブラリは1データセットだけを指定できる．

・例

```
//        EXEC  S4J
//SYSIN    DD   DISP=SHR,DSN=データセット名   ····ソースプログラム用
//INPUT    DD   DISP=SHR,DSN=データセット名   ····実行時データ用
//
```

7. S4Jの入出力エラーのメッセージ

S4J001S UNIT NUMBER u IS OUT OF RANGE(1..99).

意味： uという機番は，1～99の範囲にない（8．参照）．

システムの動作： 実行を中止する．

対策： 1～99の機番を使うようにする．

S4J002S CANNOT OPEN UNIT u,READ/WRITE MODE CONFLICT.

意味： 入出力モードに矛盾があり，機番uをオープンできない．入力用に使っていた機番をクローズせずに，出力用に使おうとしたり，その逆の場合に出るエラーである．

システムの動作： 実行を中止する．

対策： 同じ機番を入力と出力の両方に使う場合は，REWIND関数またはENDFILE関数で，いったんクローズする．

S4J003S DDNAME SNOBOOuu UNDEFIND.USE ALLOCATE COMMAND.

意味： SNOB00uuというDD名が定義されていない．

システムの動作： 実行を中止する．

対策：ALLOCコマンド等で正しくDD名を定義する．

S4J004S LRECL OF UNIT u IS TOO LONG,MUST BE < 500.

意味： 機番uのレコード長が長すぎる．

システムの動作： 実行を中止する．

対策： レコード長を500よりも小さくする．

S4J010S TOO SHORT LENGTH-SPEC.FOR UNIT u (NEDIT-HOZON FILE). SEE INPUT FUNC.

意味： 機番u（日本語保存ファイル）の最大文字数の指定が小さすぎる．

システムの動作： 実行を中止する．

対策： INPUT関数中の第三アーギュメントを大きくする（256以上）．

8．S４Jで使用するDD名

装置番号／DD名	用　　途	データセットの指定方法および条件等
1／SNOB0001 ｜ 4／SNOB0004	利用者自由	順（区分）編成の文字の入ったデータセットで，レコード長５００バイト以下．記録形式は任意．
5／SNOB0005	入力変数ＩＮＰＵＴの入力元．	Ｓ４Ｊコマンドのパラメタ ＩＮＰＵＴで指定する．無指定時は端末を割り当てる．レコード長は８０バイト以下であること．
6／SNOB0006	出力変数ＯＵＴＰＵＴの出力先．	常に端末（ＴＳＳ），あるいはＬＰ（バッチ）を割り当てる．処理系からのメッセージもここに出力する（ｉ）．
7／SNOB0007 ｜ 29／SNOB0029	利用者自由	順（区分）編成の文字の入ったデータセットで，レコード長５００バイト以下．記録形式は任意．
30／SNOB0030 ｜ 39／SNOB0039	日本語文章ファイル形式のデータセット． 入出力用．	入力時：和文エディタが作成した可変長２６０バイトのデータセットであること． 出力時：必ず可変長２６０バイトのデータセットになる．
40／－－－－－－－－	プロンプタの設定	出力用．９．３参照．
45／SNOB0045	ソースプログラムの読み込み．	Ｓ４Ｊコマンドの最初のパラメタで指定．行番号つき可変長データセットであること．最初の８０文字が処理の対象になる．
47／SNOB0047	プロファイルリストの出力先（ii）．	Ｓ４Ｊコマンドのパラメタ ＬＩＳＴで指定．無指定時は一時データセットＳ４Ｊ．ＴＥＭＰ．ＬＩＳＴを作成する．
48／SNOB0048	作業用（ii）	実行回数計数機能が使用する．
50／SNOB0050 ｜ 99／SNOB0099	利用者自由	順（区分）編成の文字の入ったデータセットで，レコード長５００バイト以下．記録形式は任意．
／FT06F001	ＦＯＲＴＲＡＮ関係のエラー情報の出力先．	常に端末（ＴＳＳ），あるいはＬＰ（バッチ）を割り当てる．

i）文法エラーのメッセージや統計情報が出力される．
ii）実行回数を計数しないときはＳＮＯＢ００４７，ＳＮＯＢ００４８を定義する必要はない．

9. ＴＳＳ環境との通信

ＴＳＳ環境下で動くプログラムの操作性をよくするためにＳ４Ｊに，次の３つの機能を追加した．

① プログラム起動時に，Ｓ４Ｊコマンドからプログラムへ任意の文字列を渡す．

② プログラム中からＴＳＳコマンドを実行する．

③ 端末からデータを入力する際に出力するプロンプタを自由に設定できる．

9．1 コマンドパラメータの受取り

①を実現するために，新たに&PARMというキーワードを追加した．たとえば，プログラムにデータセット名（NIHONGO.DATA）を渡すには，次のＳ４Ｊコマンドを入力する．

```
S4J ABC.TEXT PARM('NIHONGO.DATA') NL NP
```

この時，&PARM中には，次の文字列が入っている．

```
NL,NP,;NIHONGO.DATA
```

つまり，セミコロン（;）の後ろにPARMオペランドで指定した文字列が入っている．&PARMのセミコロンより前には，他のオペランドが次の形式ではいっている．

Ｓ４Ｊのオペランド	&PARM中での表現
LIST(dsn),L(dsn)	L,
NOLIST又はNL	NL,
PROFILE又はP	P,
NOPROFILE又はNP	NP,
SUMMARY又はS	S,
NOSUMMARY又はNS	NS,
LOW(l)又はL(l)	L=l,
HIGH(h)又はH(h)	H=h,
RESERVE(r)又はR(r)	R=r,
その他	&PARMには影響しない．

9．2 ＴＳＳコマンドの発行

②の機能を実現するために，Ｓ４Ｊ標準外部関数ライブラリ（'LIB.S4EXTLIB'）中にTSSCMDという外部関数を用意している．このTSSCMD関数を使うと，プログラムを実行中に，プログラムの中から，ＴＳＳのコマンドを実行することができる．

使用法：

```
LOAD('TSSCMD(STRING)INTEGER)')
RC = TSSCMD('TSSコマンド')
```

ただし次のコマンドは実行できない．

```
LOGON, LOGOFF, LIBRARY, PROFILE
```

？（2次メッセージ出力要求），空行

サブコマンドを必要とするコマンド（EDIT，TESTなど）

また，S4Jの使用するDD名（8．参照）と同じDD名を使用するコマンド（例えばS4J自身）は，うまく実行できない．

コマンドが正常に実行されたかどうかは，返される値（この場合ではRCの値）でわかる．

RC≧0ならコマンドが実行されたことを意味する．RCは，その時のリターンコードになっていて，通常は，0が返される．

RC＜0となるのは、次の理由でコマンドが実行されなかった場合である．

RC ＝ －4：コマンドの構文エラー
＝ －8：禁止しているコマンド
＝ －12：パラメータのエラー
＝ －20：領域確保失敗
＝ －24：アベンドエラー発生

装置機番4を端末に割り当てるためのコマンドの実行例を以下に示す．リターンコードが必要ない時には例のように'RC ＝ 'の部分を省略できる．

```
LOAD('TSSCMD(STRING)INTEGER')
TSSCMD('ALLOC F(SNOB0004) DA(*)')
```

9．3 プロンプタの設定

③を実現するために装置番号40をプロンプタの設定用とした．日本語SNOBOL4処理系は，装置番号40と結合した出力変数に代入した文字列を，通常の出力変数への代入のように直接出力するのではなく，処理系内のプロンプタ領域にいったん蓄え，端末からのデータを要求するときに改行コードを付加せずにこれを出力し，ユーザの入力を促す．処理系内のプロンプタ領域には初期値として'SNOBOL4 'という文字列が設定してある．

使用例：

```
    OUTPUT(.PROMPT, 40)
    PROMPT = 'データを入力してください '
L1  LINE = INPUT                :F(END)
    ...
                                :(L1)
END
```

この例を実行すると端末では次のように入出力が行われる（下線部が日本語ＳＮＯＢＯＬ４からの出力）．

```
データを入力してください  123 234 345
データを入力してください  A B A C A B A D A B A C A B
データを入力してください  A B A D
データを入力してください  /*
・・・
```

１０．外部関数

Ｓ４Ｊからアセンブリ言語またはFORTRAN77で書いた関数（サブルーチン）を呼ぶことができる．Ｓ４Ｊにはｓｉｎやｃｏｓのような初等関数が組み込まれてないが，FORTRAN77のライブラリと結合することにより，これらの関数を使用することができる．

１０．１　結合方法

ライブラリを結合するには，Ｓ４Ｊ実行前に，そのライブラリの入ったデータセットを適当なＤＤ名で割り当てる必要がある．

例：　ALLOC F(EXTFUNS) DA('SYS1.FORTLIB') SHR REU
　　――FORTRAN77システムライブラリをEXTFUNSというDD名で割り当てる

次に，外部関数をＳ４Ｊに組み込むためのLOAD関数の呼び出しをソースプログラム中に記述しておく必要がある．LOAD関数の呼び出し方は次の通り．

LOAD('関数名(引数1の型,引数2の型,...)結果の型','DD名')

ここで「関数名」は、組み込みたい関数の名前である．関数名とライブラリ中のメンバ名とは一致していること．「引数ｉの型」は，ｉ番目の引数の型である．「結果の型」は，関数呼び出しの結果返る値の型である．「型」としては，次の３つのみが許されている．（アセンブリ言語を使用する場合は，利用者の定義した型も許される）

型	対応するFORTRAN77の型
INTEGER	INTEGER*4
REAL	REAL*4
STRING	CHARACTER*(*)

(ただしＳＴＲＩＮＧを返す場合は，あとに示す注意が必要)

「DD名」は先ほどデータセットを割り当てた時のDD名である．

例： LOAD('SIN(REAL)REAL','EXTFUNS')

--REAL型を受け取りREAL型を返す関数SINをDD名EXTFUNSのデータセットから取り込む．

本来は，このようにするのであるが，次のような省略が可能である．まず第1に，DD名としてSNOLIBという名を用いる場合は，いちいちALLOCコマンドを使わずに，Ｓ４Ｊコマンドで

S4J ABC.TEXT LIB(ライブラリデータセット名)

と書くだけでよい．さらに，LIB(...)も省略すると，Ｓ４Ｊ標準外部関数ライブラリ（データセット名'LIB.S4EXTLIB'）が仮定される．LOAD関数の第2引数を省略した場合は，DD名SNOLIBがとられる．従って，上記の例は次のようにも書ける．

```
S4J ABC.TEXT LIB('''SYS1.FORTLIB''')
  --このように起動する．

LOAD('SIN(REAL)REAL')
  --プログラム中にはこのように書いておく．
```

１０．２ 利用者による外部関数の作成法

各利用者が独自の外部関数ライブラリを作ることもできる．ここではFORTRAN77（以下，単にFORTRANと書く）により，ライブラリをつくる方法を説明する．

結果の型がINTEGERまたはREALとなる外部関数は，FORTRANの関数副プログラムとして作成する．結果の型がSTRINGの場合は，サブルーチン副プログラムとして作成する．いずれにせよ，それらはライブラリデータセット（区分編成）中のメンバとして登録されなければならない（LINKコマンドを用

いる）．関数は「閉じている」必要がある．すなわち，未定義シンボルの参照があってはならない．

Ｓ４Ｊから引数を受け取る方法は，FORTRANから受け取る場合と変わりない．ただしＳ４ＪのSTRING型の引数を受け取る場合，注意が必要である．

例を用いて説明する．Ｓ４Ｊから外部関数ＲＥＶを次のように呼ぶとする．

```
OUTPUT = REV('ABC')
```

ここでREVはＦＯＲＴＲＡＮで次のように書いてあったとしよう．

```
      SUBROUTINE REV(S)
      CHARACTER S*(*)
      INTEGER L,LEN              --LENはFORTRAN組込関数。
      L=LEN(S)                   --LにSの「長さ」を得る。
           .
           .
           .
      RETURN
      END
```

このとき，Ｌに入る値は，渡された文字型変数の文字数であって，バイト数ではない．バイト数はこの数の２倍である．これはＳ４Ｊの内部での文字表現が，日本語文字・EBCDIC文字にかかわらず，１文字２バイトであることに起因する．すなわち，EBCDIC文字１文字は，第１バイト目に０，第２バイト目に該当するEBCDICコード，という形で表現されている．日本語文字の場合は，該当するＪＥＦコード（２バイト）がそのまま入っている．

例：

00	C1	00	C2	00	C3

EBCDIC文字 ABC

A3	C1	A3	C2	A3	C3

日本語文字　　ＡＢＣ

これをFORTRANから参照する場合，FORTRANの２文字でＳ４Ｊの１文字を参照することになる．上のプログラムでは，部分文字列

Ｓ（１：２）　が第１文字　Ａ

Ｓ（３：４）　が第２文字　Ｂ

Ｓ（５：６）　が第３文字　Ｃ

を参照する．つまり一般に，

λ=LEN(σ)

ならば，文字型変数σの中で実際にデータが入っているのは，

σ(1:2*λ)

ということになる．

FORTRANからINTEGER型またはREAL型の値を返す方法は，通常の関数副プログラムからの返し方と何ら変わらない．STRING型を返すには（S４Jから直接呼ばれたサブルーチン副プログラムの中で），次のCALL文を実行すればよい．

```
CALL RTNSTR(σ,λ)
```

ここで，σは文字型変数で，返すべき文字列を設定しておく．λは整数型（4バイト）変数でσの文字数を設定しておく，特別なサブルーチンRTNSTRは，ライブラリデータセット'LIB.S4EXTLIB'中にあり，LINKコマンドで組み込むこと．（ＬＩＢオペランドに指定する．）

S４Jへ「失敗」を知らせるには，次のRETURN文で戻ればいい．

```
RETURN 1
```

１０．３ 外部関数の作成例

以下に，外部関数の作成例を示す．例にとりあげる関数REVは，S４JからSTRING型の引数を受け取り，文字の順序を逆にしてS４Jへ返す，というものである．扱える文字列の長さは２５６文字までとし，それを超える文字列を受け取った場合は「失敗」とする．

FORTRAN77のプログラム（データセット名 REV.FORT77）

```
      SUBROUTINE REV(S)
      CHARACTER S*(*)
      CHARACTER BUF*512
      INTEGER I,L,LEN
      L=LEN(S)
      IF(L.GT.256)RETURN 1
      DO 10 I=1,L*2-1,2
   10  BUF(I:I+1)=S(L*2-I:L*2-I+1)
      CALL RTNSTR(BUF,L)
      END
```

コンパイルとリンクのためのコマンド

```
FORT77 REV OBJ(REV) NAME NOGO
```

--REV.FORT77をコンパイルしてオブジェクトファイルREV.OBJをつくる．

```
LINK REV.OBJ LO(MYS4LIB) F LIB('LIB.S4EXTLIB')
```

--REV.OBJにFORTRAN実行時ルーチン，RTNSTRサブルーチンとを結合して，ロードモジュール（ライブラリ）MYS4LIB.LOAD(REV)をつくる．

S4Jのプログラム（データセット名REVTEST.TEXT）

```
      LOAD('REV(STRING)STRING')
      OUTPUT = REV('これはさかさま')
    END
```

これをコマンド

```
    S4J REVTEST.TEXT LIB(MYS4LIB.LOAD)
```

で実行させると，

　　まさかさはれこ

という出力が得られる．

11．日本語ラインプリンタへの出力

S4Jから日本語ラインプリンタへ出力する方法は，普通のファイルへの出力と同様であるが，実行前に（あるいはTSSCMD関数（9．2参照）で，実行中に）次のTSSコマンドを入力しておく必要がある．（バッチの場合にはこれに相当するDD文をジョブデック中に入れておく．）

```
ATTR NLP OUTPUT RECFM(A)

ALLOC F(SNOBOOuu) SYS(S) US(NLP) REU
```

この後，S4Jから機番uuへの出力を行えば，それはラインプリンタへ送られる．

ラインプリンタへ文字を送る場合，各行の第1文字目には「制御文字」を書く必要がある．この制御文字の付加は，

```
    NLP = ' ' BUF
```

のように、出力のたびに行ってもいいが，OUTPUT関数であらかじめ指定することもできる．

```
    OUTPUT(.NLP,10,' ')
```

としてから

```
    NLP = BUF
```

とするのと，前の例とは同じ動作をする．

日本語ラインプリンタでは，特定の文字（空白とアンダライン）を除いては，重ね打ちができないことに注意する必要がある．

参考文献

1．武富，高木，川崎他　日本語情報システムJEFの使用法，九州大学大型計算機センター広報，13，4，1980，406－468．

2. L. Cherry Computer Aids for Writers, ACM SIGPLAN NOTICES, **16**, 6, 1981, 61−67.

3. R. E. Griswold, J. F. Poage, I. P. Polonsky THE SNOBOL 4 PROGRAMMING LANGUAGE, Prentice−Hall, Inc., 1968.

4. 角田 SNOBOL, 情報処理, **22**, 6, 1981, 473−476.

5. 牛島, 黒坂, 日並, 栗山 JEF日本語処理機能を追加したSNOBOL4について, 九大大型計算機センター広報, **16**, 2, 1983, 155−179.

6. 吉田, 牛島 日本語SNOBOL4におけるパターンマッチ処理の性能向上について, 情報処理学会第27回全国大会論文集 (**6E-8**), 1983, 423−424.

7. 吉田, 牛島 日本語SNOBOL4処理系のためのIOプリミティブの設計と実現, 情報処理学会第28回全国大会論文集 (**2H-7**), 1984, 355−356.

8. 計算機マニュアル FDMS (和文エディタ) ／JEF解説書, **64AR-8211**, 富士通 (株).

9. 牛島, 高比良 SNOBOL4の使用について, 九大大型計算機センター広報, **12**, 1, 1979, 5−19.

10. J. F. Gimpel Algorithms in SNOBOL4, John Wiley & Sons, 1976.