

Title	ACOS1000プログラミング言語の使用手引（第1版）
Author(s)	後藤, 米子
Citation	大阪大学大型計算機センターニュース. 1984, 54, p. 53-70
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/65619">https://hdl.handle.net/11094/65619</a>
rights	
Note	

*Osaka University Knowledge Archive : OUKA*

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

Osaka University

# ACOSシステム1000プログラミング言語の使用手引

( 第 1 版 )

研究開発部 後 藤 米 子

## 1. はじめに

本稿の目的は、昭和59年8月現在の、大阪大学大型計算機センターのACOSシステム1000における、プログラミング言語の操作法の入口を解説することである。文法に関することには一切触れないので、マニュアルや他の教科書を参考にさせていただきたい。このマニュアルとは、日本電気(株)が発行している市販の仕様書のことである。日本電気提供のプログラミング言語でも、ジョブ制御言語 ( Job Control Language ; 以下 JCL と略す ) やインタプリタを起動するためのコマンド名は、マニュアルの記述とは異なるものもあるので、本稿を参照していただきたい。

現在、本センターで使用できるプログラミング言語は、すでに10種類を越えている。2.では、これらの概観を行い、3.では、具体的な処理形態などの一覧を示す。4.以降で個別のプログラミング言語の使用法について説明する。ACOSシステム1000のアセンブラ言語であるGMAPと、同じくシステム記述言語であるHPL、および利用者のきわめて少ないものについては今回は省略した。理解を助けるために、一部のプログラミング言語の例題と、そのコマンドファイルおよびJCLをファイルに登録している。使用する際には、これらを活用していただきたい。

## 2. 概観

### 2.1 処理形態と処理モード

各プログラミング言語によって、使用できる処理形態および処理モードが定められていることに注意されたい。処理形態には、バッチ処理とTSS処理がある。ここで言う処理モードとは、計算機内部におけるプログラムの実行時のモードのことであり、実空間のみを使用するRモードと、仮想空間を使用するVモード<sup>24)</sup>がある。ACOSシステムの当初はRモードのみであったが、後にVモードがFortran言語を始めとして開発された。このような背景から、現在、Vモードが使用できるものは表1のように限られている。C, PASCAL, LISP 1.9などは、Rモードのみであることに留意されたい。Rモードではプログラムが実空間でのみ実行されるため、アーキテクチャの制約からプログラム・サイズの最大は、バッチ処理では240 K語、TSS処理では210 K語となる。Vモードでは仮想空間が使用される。その最大はジョブクラスごとに異なり、バッチ処理では、2000 K語 ( Eクラス ) から16000 K語 ( Cクラス )、TSS処理では、4000 K語である。

両モードで使用可能なものは、コンパイラ・オプションにその処理モードを指定して区別する。RモードとVモードはそれぞれ異なる命令の体系を採用しており、これらは一つの実行可能プログラムのなかで混在できない。

## 2.2 コンパイラ・オプションの既定値

コンパイラは種々の機能を持っているが、運用の既定値ではこれらの機能のうち、必要な最小限を採用している。コンパイラ・オプションの既定値はマニュアルのとおりである。ただし、資源節約のためバッチ処理では、ソースプログラムリストを出力しないNLSTINに変更されている。

実行時の効率をあげるために以下のものは最適化の機能をもつが、運用の既定値では休止状態にある。

C, Fortran 77(V), Fortran(R), IBM互換のPL/1, 標準PL/1.

コンパイラ・オプションの指定の順序は任意でよい。

## 2.3 JIPS コード

JIPSコード(日本電気の漢字コードの名称)<sup>19,20)</sup>が使用できるものは、次のとおりである。

Cobol 74, Cobol, Fortran 77(V), IBM互換のPL/1, ShapeUp.

## 2.4 エスケープシーケンス

C, PASCAL, APL, ShapeUpでは、構文を記述するために特殊文字が使用されている。これらの特殊文字が端末のキー・ボードにない場合に代用するための文字列が、エスケープシーケンスである。

これらはプログラミング言語によって固有に定まっている。マニュアルなどの資料には明記されているので、それを参照していただきたい。

## 2.5 本稿の表記に関する補足事項

本稿では、RモードおよびVモードの双方で使用できるもののうち、その区別を要するものは言語名の後に括弧で囲みモードを示した。PL/1には後述のように2種類あるので、誤解のないようにそれぞれIBM互換のPL/1, 標準PL/1とよぶことにする。

4以降のJCLの、'000000000'と、'password'には利用者自身の課題番号とパスワードを指定し、'job-class'にはE,A,B,Cのいずれかのジョブクラスを指定しなければならない。個別のJCLに関する説明は省略する。バッチ処理のRモードでは、プログラム・サイズの既定値は120K語であるため、必要に応じて\$LIMITS文で拡張しなければならない。CPU時間はジョブクラスにかかわらず、既定値は36秒である。JCLファイルは、CARDINサブシステムで実行するものと仮定し、先頭にその応答を記述しているので、RUNコマンドのみをタイプインすればよい。

TSS処理の使用例の下線は、利用者の入力であることを示す。ソースプログラムファイル、コ

マンドファイルまたはJCLファイルアクセスする場合は、リードパーミッション(ファイル名の直後に文字列, Rを指定する)が必要である。これらについては、TSSの手引<sup>21)</sup>を参考にしてください。[ ]は、括弧内の文字列を省略できることを示す。

### 3. 一 覧

本センターで利用可能なプログラミング言語の種類とその処理形態、処理モードおよび準拠規格などの一覧を表1に示す。表の見方は次のとおりである。処理形態の欄の「○」および「×」は、それぞれの処理形態で可能および不可能であることを示し、処理モードの欄の「R」および「V」は、それぞれ、RモードおよびVモードで使用できることを示す。下線は、両モードで使用可能なもののうち運用の標準を示す。参考文献の欄のうち、日本電気提供のものはマニュアルの分類番号である。

表 1. プログラミング言語一覧

プログラミング 言 語 名	処 理 形 態		処 理 モ ー ド	提 供	参 考 文 献	備 考
	バッチ	TSS				
ALGOL	○	○	R	日本電気	FGC02, 03	JIS 5050 準拠
APL	×	○	R	日本電気	FGL01	IBM VS APL 互換
BASIC	×	○	R	日本電気	FGH01	
C	○	○	R	日本電気	FGP01	
COBOL74	○	○	<u>R</u> , V	日本電気	FGA05, 06	JIS C 6205 準拠
COBOL	○	×	R	日本電気	FGA02, 04	
FORTRAN77	○	○	R, <u>V</u>	日本電気	AGB01, FGB8, 9	JIS C 6201 準拠
FORTRAN	○	○	<u>R</u> , V	日本電気	FGB2, 3, 4, 5	旧JIS 水準 7000
FORTRAN E	○	○	R	日本電気	FGB06	実行型FORTRAN
GMAP	○	○	<u>R</u> , V	日本電気	FGE01, 02	ACOS アセンブラ
HPL	○	○	<u>R</u> , V	日本電気	FGN01, 02	ACOS システム 記述用
LISP 1.5	○	○	R	日本電気	FGM01	
LISP 1.9	×	○	R	電 総 研	4)	
PASCAL	○	○	R	日本電気	FGK01	
PL/I	○	○	<u>R</u> , V	日本電気	FGD04, 05	IBM 互換
PL/I	○	○	R	日本電気	FGD 02, 03	ISO 標準
Reduce	○	○	R	東京大学	2)	
ShapeUp	×	○	R	日本電気	講習会テキスト	
SLISP	○	○	R	東北大学		
Westran	×	○	R	電 総 研		出力はFORTRAN
XXPL	×	○	R	電 総 研	3)	コンパイラ記述用

#### 4. ALGOL, Cobol, Cobol 74 (R), IBM 互換のPL/1 (R), 標準PL/1 のジョブ制御言語

ALGOL (ALGOrithmic Language の略) は、アルゴリズムを正確に記述するための言語として開発されたものである。ALGOL には仕様の異なる ALGOL 60 と ALGOL 68 があり、ALGOL 68 のほうが言語としては新しい。本センターで使用できるものは、JIS 規格に準拠した ALGOL 60 に対応するものである。ALGOL 68 に対応した JIS 規格はない。

Cobol (COmmon Business Oriented Language の略) は、事務処理のための言語として開発されたものである。本センターで使用できるものには、Cobol 74 と Cobol の 2 つの処理系があるが、前者のほうが機能が大きく、バッチ処理および TSS 処理ともに V モードが使用できる。本稿では V モードに関する説明は省略する。

PL/1 (Programming Language One の略) は、アルゴリズムやシステムを記述するために IBM 社で開発された言語である。本センターで使用できるものには、IBM 互換の PL/1 と ISO 標準規格の PL/1 (標準 PL/1 とよぶ) の 2 つがあり、両者の区別はコンパイラ・オプションで行う。運用の既定値では IBM 互換の PL/1 が動作する。機能の拡充は IBM 互換の PL/1 に重点がおかれており、最近 V モードが使用可能になった。

##### 4.1 バッチ処理

ジョブの基本 JCL の構成は次のとおりである。使用するプログラミング言語に応じて、パラメータ language に表 2 のものを指定する。

	1	8	16
1.	\$	JOB	000000000\$password, job-class
2.	\$	SETUP	language
3.	\$		language
4.			ソースプログラム
5.	\$	GO	
6.	\$	LIMITS	ttt, m1, m2, ppppp
7.			データ
8.	\$	ENDJOB	

表 2. パラメータ language の値およびサブシステム名

プログラミング言語名	language	サブシステム名
ALGOL	ALGOL	ALGOL
COBOL 74	CBL 74	CBL 74
COBOL	COBOL	—
IBM 互換の PL/1	PL 1	PL 1
標準 PL/1	PL 1	PL 1

## 4.2 TSS処理

システム選択レベル (SYSTEM?) またはビルドモード (\*) で、表2のサブシステム名をタイプインする。サブシステム名は先頭4文字で識別されるので、先頭の4文字だけをタイプインしてもよい。これらのサブシステムのもとではプログラムの実行だけでなく、新規にソースプログラムを作成したり既存のものを修正するなどの編集処理を行うことができる。コンパイラを起動するにはRUNコマンドを使用する。RUNコマンドの形式は、言語により異なる。

### (1) ALGOL のRUNコマンドの形式

---

RUN-n fs = fh ; fc(option) ulib#fe

---

各パラメータの意味は次のとおりである。=, ;, ( ), #は各パラメータ間の区切り記号である。

-n 実行時のCPU時間を秒単位で指定する(省略すると900秒となる)。

fs 入力となるファイル記述子を指定する(カレント・ファイルの場合は\*)。入力ファイルがカレントファイルのみの場合は、省略してもよい。複数個の入力ファイルを指定する場合は、一つのセミコロンで区切る。

fh 実行形式プログラムファイルを保存するときにその出力用のファイル記述子を指定する。

fc 相対形式プログラムファイルを保存するときにその出力用のファイル記述子を指定する。

option コンパイラ・オプション(言語で異なる)およびローダ・オプションを指定する。複数個のオプションを指定する場合は、一つのコンマで区切る。

ulib 利用者ライブラリを使用するときに指定する。

fe 実行時に必要なファイルとファイルコードの対応を指定する。

RUNコマンドの代わりにRUNHコマンドを使用すると、プログラムの実行に先立ちこのRUNコマンドが投入された時刻が端末に直ちに表示される。

例-1 ソースプログラムにはパーマメントファイルFILESを指定し、実行時のデータはパーマメントファイルDATAから機番12で入力する。

SYSTEM? ALGOL N

\* RUNH FILES#DATA "12"

### (2) Cobol 74(R), IBM 互換のPL/1(R), 標準PL/1のRUNコマンドの形式

RUNコマンドのパラメータは、入力ファイルを指定する欄とオプションを指定する欄があり両者の間を一つのコロンの(:)で区切る。入力ファイルまたはオプションを複数個指定する場合は、それぞれのオプションの間を一つの空白で区切る。

---

RUN fs:option

---

fs : 入力となるファイル記述子を指定する。入力ファイルがカレントファイルのみの場合は、省略してもよい。

option : コンパイラおよびローダに対するオプションを指定する。

以下にTSS処理固有のオプションの一部を簡単に紹介する。キーワードの文字列中の小文字は、省略できることを示す。

File = ファイル記述子 [, パーミッション] (fc)

実行時に必要なファイルとファイルコードの対応を指定する。

Library = ファイル記述子

ライブラリ (5.4 参照) を使用するとき指定する。

DECK = ファイル記述子

相対形式プログラムファイルを保存するときその出力用のファイルを指定する。

NOLoad 翻訳処理のみを行うときに指定する。

TIME = n 実行時に必要なCPU時間を秒単位で指定する (省略すると900秒となる)。

Hstar = ファイル記述子

実行形式プログラムファイルを保存するときその出力用のファイルを指定する。

NoGo 翻訳処理および結合編集処理を行うときに指定する。

#### 4.3 使用例

例-1 IBM互換のPL/1でソースプログラムS1およびS2を、最適化オプションで実行する。  
実行時のCPU時間の制限は60秒と指定する。

SYSTEM? PL1 N

\* RUNH S1 S2:OPTZ TIME=60

例-2 ソースプログラムS1を修正しS2と結合して実行する。

SYSTEM? PL1

OLD OR NEW-OLD S1

修正作業

\* RUNH \* S2:OPTZ またはRESAVE S1の後、RUNH S1 S2:OPTZ

でもよい。

IBM互換のPL/1のソースプログラムとJCLを次のファイルに登録している。

EXAMPLE/GENGO/PL1/S1 (ソースプログラム)

EXAMPLE/GENGO/PL1/J1 (上記プログラムのJCL)

## 5. Fortran 77 (V)

Fortran (FORmula TRANslation の略) は、数値計算のための言語である。本センターの Fortran 言語の標準は、Fortran 77(V)<sup>12~14)</sup> であり、5.4 に示すように、数値計算ライブラリや図形ライブラリが用意されている。実行時の高速化をはかるために、IAP(Integrated Array Processor の略)<sup>23)</sup> を使用することができる。コンパイラ・オプションの既定値は次のとおりである。ただしソースプログラムの形式に関しては、バッチ処理と TSS 処理で異なるので除外している。下線はバッチ処理の既定値でマニュアルと異なるものである。

ALC, ALIGN, NASTER, AUTODBL=NONE, BYNAME, NCOMDK, NDECK, NDIAG, NDLR, EJECT, NFLTCHK, NRVSET, NIAP, NINCLST, NINCLUDE, INLINE=0, JREST, NLSTIN, NLSTOU, NOMAP, MAXLIN, (54), OPT=1, REST, NROUND, STATIS, NUNF, NXREF.

### 5.1 バッチ処理

	1	8	16
1.	\$	JOB	0000000000 \$password, job-class
2.	\$	FRT77	オプション
3.		ソースプログラム	
4.	\$	GO	オプション
5.	\$	LIMITS	ttt,,,ppppp
6.		データ	
7.	\$	ENDJOB	

\$GO 文のオプションについては 5.4 参照。プログラム・サイズの指定は不要である。

### 5.2 TSS 処理

システム選択レベル (SYSTEM?) またはビルドモード (\*) で、サブシステム名 FRT77 (FRT7 でもよい) をタイプインする。このサブシステムのもとではプログラムの実行だけでなく、新規にソースプログラムを作成したり既存のものを修正するなどの編集処理を行うことができる。コンパイラを起動するには RUN コマンドを使用する。RUN コマンドのパラメータは、入力ファイルを指定する欄とオプションを指定する欄の二つがあり、この間を一つのコロン (:) で区切る。入力ファイルまたはオプションを複数個指定する場合は、一つの空白で区切る。

RUN fs:option

fs : 入力となるファイル記述子を指定する。入力ファイルがカレントファイルのみの場合は省略してもよい。

option : コンパイラまたはリンカに対するオプションを指定する。



以下にTSS処理固有のオプションの一部を簡単に紹介する。キーワードの文字列中の小文字は、省略できることを示す。

File=ファイル記述子[, パーミッション](fc)

実行時に必要なファイルとファイルコードの対応を指定する。

Library=ファイル記述子

ライブラリ(5.4参照)を使用するときに指定する。

TERMinal = fc

装置番号 5, 6, 41, 42, 以外で端末入出力をおこなうときに指定する。

DECK = ファイル記述子

相対形式プログラムファイルを保存するときにその出力用のファイルを指定する。

NoLoad= 翻訳処理のみを行うときに指定する。

TIME=nn 実行時に必要なCPU時間を秒単位で指定する(省略すると900秒となる)。

RU = ファイル記述子

実行形式プログラムファイルを保存するときにその出力用のファイルを指定する。

NoGo = 翻訳処理および結合編集処理を行うときに指定する。

RUNコマンドの代わりにRUNHコマンドを使用すると、プログラムの実行に先立ちこのRUNコマンドが投入された時刻が端末に直ちに表示される。

### 5.3 使用例

例-1 カレントファイルのソースプログラムを実行する。‘READ(1, . . . . .’によるデータ入力はパーマネントファイルDATAから行う。入力ファイルであるため、パーミッションをRと指定する。

```
SYSTEM? FRT77 O EXAMPLE
```

```
* RUN : F=DATA, R(01)
```

例-2 カレントファイルのソースプログラムを数値計算ライブラリASL<sup>18)</sup>を使用して実行する。

```
SYSTEM? FRT77 O EXAMPLE 2
```

```
* RUN:L=LIB/ASL
```

### 5.4 ライブラリ

Fortran 77(V) から使用できるライブラリには、表3に示すものがある。これらの使用方法は次のとおりである。

バッチ処理：\$GO文のオプションで指定する。

TSS処理：RUNコマンドのオプションL=で、表3のライブラリファイル名を指定する。表の左の欄の数字は、次のものを示す。

1. 数値計算ライブラリ 2. 図形ライブラリ 3. 最小二乗法<sup>22)</sup>

コンパイラ・オプションは、運用の標準と異なるものだけを示している。オプションASMDARYは、引用するプログラム単位にもその指定が必要である。オプションCOMPATは、引数に文字型を使用するときに指定する。

表3. ライブラリおよびアプリケーション一覧

	ライブラリ名	\$GOオプション	ライブラリ・ファイル名	コンパイラ・オプション
1	ASL	ASL	LIB/ASL7	OPT=2, INLINE=2, IAP
	MATHLIB-6	MLIB	LIB/MLIB7	上記+COMPAT
	センター・ライブラリ	CLIB	LIB/CLIB7	OPT=2, INLINE=1
2	GDSP-6/PLOT	—	LIB/GPLTV	} OPT=2, INLINE=2, COMPAT, ASMDARY
	GDSP-6/CPLLOT	—	LIB/CPLTV	
	GDSP-6/SPLLOT	—	LIB/SPLTV	
	統合化ライブラリ	XLIB XLIBS	LIB/AVLIBと 各種デバイス・ルーティン	COMPAT
3	SALS			} OPT=2, INLINE=1
	• 単精度版	SLIB	LIB/SLIB7	
	• 倍精度版	SLIBD	LIB/SLIB7D	

## 6. Fortran (R)

旧 JIS 規格であり本センターの運用では非標準の Fortran 言語である。Fortran 77(V) (5 参照) のほうが機能もはるかに大きく本センターの運用の標準となっているのでできるだけ Fortran 77(V) を使用していただきたい。

### 6.1 バッチ処理

	1	8	16
1.	\$	JOB	000000000 \$password, job-class
2.	\$	FORTAN	オプション
3.			ソースプログラム
4.	\$	GO	FORT
5.	\$	LIMITS	ttt,,,ppppp
6.			データ
7.	\$	ENDJOB	

## 6.2 TSS処理

システム選択レベル(SYSTEM?)またはビルドモード(\*)で、サブシステム名 FORTRAN (FORTでもよい)をタイプインする。

---

RUN-n fs=fh;fc(option)ulib#fe

---

各パラメータの意味は次のとおりである。=, ;, ( ), #は各パラメータ間の区切り記号である。

-n 実行時のCPU時間を秒単位で指定する(省略すると900秒となる)。

fs 入力となるファイル記述子を指定する(カレント・ファイルの場合は\*)。入力ファイルがカレントファイルのみの場合は、省略してもよい。複数個の入力ファイルを指定する場合は、一つのセミコロンで区切る。

fh 実行形式プログラムファイルを保存するときにその出力用のファイル記述子を指定する。

fc 相対形式プログラムファイルを保存するときにその出力用のファイル記述子を指定する。

option コンパイラ・オプションおよびローダ・オプションを指定する。複数個のオプションを指定する場合は、一つのコンマで区切る。

ulib 利用者ライブラリを使用するときに指定する。

fe 実行時に必要なファイルとファイルコードの対応を指定する。

RUNコマンドの代わりにRUNHコマンドを使用すると、プログラムの実行に先立ちこのRUNコマンドが投入された時刻が端末に直ちに表示される。

## 7. APL

APL(A Programming Languageの略)は、K.E. Iversonによってアルゴリズムを論理的に簡潔に記述するための表記法として考え出された言語である。処理方式はインタプリタであり、会話的に実行される。

### 7.1 使用法

APLを使用するときには通常はAPL専用の端末(APL端末と呼ぶ)を使用する。この端末は、キー・ボードにAPLで使用する特殊記号(演算子)の文字をすべて備えている。APLはAPL端末以外にJIS端末やASCII端末からも使用できる。このときにはエスケープシーケンスを使用しなければならない。

APL端末では端末の接続は、\$\$\$CON, TSS, , APLとタイプインする。端末の接続方法についてはTSSの手引を参照していただきたい。システム選択レベル(SYSTEM?)でAPLとタイプインすると、APL端末以外の場合は端末タイプの入力を求められるので、そのタイプに応じて次のものをタイプインする。

ASCII 端末の場合は、ASCII

JIS 端末の場合は、UASCII

一度サブシステムが呼ばれると、このなかではEDITOR サブシステムなどを使用することができずAPLで許される文またはデータだけを入力できる。APLサブシステムのなかでは、作業空間(ワークスペース)を介して実行を行い、これをパーマネントファイルに保存しておくことができる。サブシステムを終了するときには、) DONEをタイプインする。

## 7.2 使用例

マニュアルから例を抜粋しサブシステムの起動を含めて、次のコマンドファイルに登録している。これを実行するにはシステム選択レベル(SYSTEM?)でこのファイルを入力ファイルとするCRUNコマンドをタイプインするとよい。

EXAMPLE/GENGO/APL/C-J(JIS 端末用、マニュアルFGL 01 P.36 図4-51 参照)

## 8. BASIC

BASIC(Beginner's All-purpose Symbolic Instruction Codeの略)は計算機の初心者でも容易に修得できることを目的に作成された言語である。JIS規格(JIS C 6207)が1982年に制定されており、処理系はこれに準拠している。このことはマニュアルには記載されていない。

TSS処理でのみ使用でき、処理方式はコンパイル & 実行の方式であるため、オブジェクトプログラムを保存したりすることはできない。システム選択レベル(SYSTEM?)またはビルドモード(\*)でサブシステム名BASIC(BASIでもよい)をタイプインする。実行コマンドは、AIGOLとほぼ同じであるので、4.2の(1)を参照していただきたい。BASICサブシステムのもとでは、編集処理を行うことができる。サブシステムを終了するときには、DONEをタイプインする。

## 9. C

言語Cはシステム記述用言語である。B.W.KernighanとR.M.Ritchie共著の'The C PROGRAMMING LANGUAGE'に準拠しており、TSS処理およびバッチ処理で使用できる。言語Cの一般的な特長などについては、すでに適切な解説記事<sup>1)</sup>がセンターニュースに掲載されているので、それを参照していただきたい。ソースプログラムの文字列には、英大文字および英小文字のいずれも使用でき、バッチ処理の既定値では区別されないが、TSS処理では英小文字が既定値となっているので、英大文字を使用するときにはコンパイラ・オプション-Aが必須である。Cのマニュアルには、エラーメッセージの解説は掲載されていないので、リストに表示されている英文のエラーメッセージテキストを判読する以外に方法がない。コンパイラは最適化の機能をもつ。

B言語は現在使用できなくなっているが、それにかわるものとしてもCを使用していただきたい。

## 9.1 バッチ処理

	1	8	16
1.	\$	JOB	0000000000 \$password, job-class
2.	\$	BREAK	
3.	\$	LOWLOAD	
4.	\$	OPTION	NOMAP
5.	\$	CC	
5.		ソースプログラム	
6.	\$	LIBRARY	CC, BF
7.	\$	EXECUTE	
8.	\$	LIMITS	ttt, m1, m2, ppppp
9.	\$	PRMFL	CC, R, R, LIB/C/LIB1
10.	\$	PRMFL	BF, R, R, LIB/C/LIB2
11.		データ	
12.	\$	ENDJOB	

6.の“CC”と“BF”は必ずこの順序で指定しなければならない。この順にライブラリの検索が行われるからである。9.および10.のファイル名はマニュアルと同一の指定でもよい。

## 9.2 TSS処理

システム選択レベル(SYSTEM?)またはビルドモード(\*)でCCコマンドをタイプイン後、次の入力促進(CCコマンドをSYSTEM?でタイプインしたのであればこの場合もSYSTEM?となる)に対してGOコマンドをタイプインする。CCコマンドは、指定されたファイルをソースプログラムとして処理し、実行形式プログラム(既定値ではテンポラリファイルでファイル名.H)を作成する。実行に至らないことに注意されたい。GOコマンドは、指定された実行形式プログラムファイルから実行の処理を行う。これらのコマンドは処理を起動するためのものであり、ソースプログラムの作成や修正などの編集処理は、EDITORサブシステム(FRT77サブシステムを選択してもよい)のもとで行う必要がある。TSS処理では実行時のライブラリは自動的に結合される。

コマンドの形式を次に示す。どのパラメータも、複数個を指定するときの区切り記号は、一つの空白である。オプションの順序は任意であり、大文字・小文字の区別はない。

---

CC コマンドの形式    CC filename option

---

filename : ソースプログラムのファイル記述子を指定する。カレントファイルの場合は \* を指定する(省略不可)。

option : 以下にその一部を示す。( )内はその意味である。

- A (大文字・小文字の区別をしない)      - O (最適化機能を使用)

- Hfin (ファイル名 fin の実行形式プログラムを作成)

JIS 端末でソースプログラムを作成した場合は -A は必須である。

---

GO コマンドの形式    GO : filename option

---

filename : 実行形式プログラムの登録されているファイル記述子を指定する(.Hが既定値)。

option : Cプログラムに渡すアーギュメント、または実行時に必要なファイルの再割り当てを行う。

実行時のCPU時間を指定するオプションがないので、無限ループに陥らないように注意しなければならない。

### 9.3 エスケープシーケンス

次に示す7つの文字は、矢印の右側のエスケープシーケンスをもつ。

{ → \$( , } → \$ ) , [ → \$ < , ] → \$ > , / → \$ - , | → \$ + , ~ → \$ ! .

### 9.4 使用例

例 ファイルFILEINからソースプログラムを入力し最適化機能を使用する。実行形式プログラムをテンポラリファイルPPに登録後直ちにこれを実行する。FILEINには大文字を使用している。

```
SYSTEM? CC FILEIN -A -O -HPP
```

```
SYSTEM? GO PP
```

マニュアルから例を抜粋し、下記のファイルにバッチ処理用のJCLとTSS処理用のコマンドファイルに登録している。コマンドファイルを実行するには、SYSTEM?でこのファイルを入力ファイルとするCRUNコマンドをタイプインする。

```
EXAMPLE /GENGO /C /C1 (コマンドファイル)
```

```
EXAMPLE /GENGO /C /J1 (JCL)
```

ソースプログラムは、マニュアルFGP01のp. 79を参照していただきたい。

## 10. LISP 1.5 および LISP 1.9

LISP (LIST Processor の略)は、リスト処理用のプログラミング言語である。本センターで使用できるLISPにはLISP 1.5とLISP 1.9がある。いずれもインタプリタとコンパイラの2つの処理方式をもち、プログラムが長くなるとコンパイラを使用するほうが実行時のCPU時間が短くてすむ。LISP 1.5は日本電気提供のシステム標準のものであり、TSS処理とバッチ処

理で使用できるが、バッチ処理での使用法は省略する。LISP 1.9は電総研から譲渡されたものでTSS処理で使用できる。LISP 1.9の名称は、電総研より譲渡された当初の呼称を現在も採用しており、現バージョンは、LISP 2.1と称されているものと同一である。

### 10.1 LISP1.5の使用法

システム選択レベル (SYSTEM?) またはビルドモード (\*) で \$L 1.5 をタイプインすると  
COMPILE?

の問合せがあるので、インタプリタを使用するときにはNO、コンパイラを使用するときにはYESを入力する。

使用例としてマニュアルから次の2つの例を抜粋し、それぞれ次のコマンドファイルに登録している。実行するにはSYSTEM? またはビルドモードでこのファイルを入力ファイルとするCRUNコマンドをタイプインする。

インタプリタの使用例 EXAMPLE/GENGO/LISP1.5/C1 (マニュアルFGM01 p. 64 図 7.2参照)

コンパイラの使用例 EXAMPLE/GENGO/LISP1.5/C2 (マニュアルFGM01 p. 38 図 4.1参照)

### 10.2 LISP1.9の使用法

システム選択レベル (SYSTEM?) またはビルドモード (\*) で \$L 1.9 とタイプインすると、データ領域管理に関して

STANDARD?

と問合せがあるので既定値で実行するときはキャリジ・リターン の応答をする。続いて

RUN

とタイプインするとインタプリタが実行される。このレベルでは入力促進記号は? である。

\$L 1.9 のコマンド入力レベル (\* が入力促進記号) では、編集処理をおこなうためにEDITOR サブシステムを使用できる。

使用例として仕用書から例を抜粋し、次のコマンドファイルに登録している。これを実行するには、システム選択レベル (SYSTEM?) またはビルドモードでCRUNコマンドをタイプインする。

インタプリタの使用例 EXAMPLE/GENGO/LISP 1.9 /C1 (参考文献 7) 1-5, 1-6 参照)

## 11. PASCAL

PASCALはアルゴリズムの記述、プログラミングの教育などに幅広く使用されている言語である。本センターで使用できるPASCALはK. JensenとN. Wirthの共著の‘Pascal User Manual and Report’第2版に準拠しており、TSS処理およびバッチ処理で使用できる。

## 11.1 バッチ処理

	1	8	16
1.	\$	JOB	0000000000 \$password, job-class
2.	\$	BREAK	
3.	\$	LOWLOAD	
4.	\$	OPTION	NOMAP
5.	\$	PASCAL	
6.		ソースプログラム	
7.	\$	SELECT	CMDLIB/PASCAL/EXECUTE
8.		データ	
9.	\$	ENDJOB	

7.の \$SELECT文は、ファイルに登録されている JCL を指定する。このファイルの内容はLIST コマンドでみることができるのでその通りに指定してもよいが、\$SELECT文を使用するほうが簡便であろう。

## 11.2 TSS処理

システム選択レベル(SYSTEM?)またはビルドモード(\*)でPASCAL(PASCでもよい)コマンドをタイプインする。PASCALコマンドはソースプログラムの翻訳処理から実行までを行う。実行形式プログラムを作成しておきこれから実行することもできるが、この場合はGO コマンドを使用する。コマンドの形式は次のとおりである。どのパラメータも複数個を指定するときには、一つの空白で区切る。

---

PASCAL コマンドの形式	PASCAL filename option
----------------	------------------------

---

filename : 入力となるファイル記述子を指定する。

option : コンパイラまたはローダに対するオプションを指定する。

---

GOコマンドの形式	GO : filename option
-----------	----------------------

---

filename : 実行形式プログラムの保存されているファイル記述子を指定する。

option : PASCAL プログラムに渡すアークギュメント、または実行時に必要なファイルの再割り当てを行う。

実行時のCPU時間を指定するオプションがないので、無限ループに陥らないように注意しなければならない。



### 11.3 エスケープシーケンス

次の5文字は矢印の右側のエスケープシーケンスをもつ。

{ → ( \* , } → ) \* , [ → ( . , ] → ) . , / → @ .

### 11.4 使用例

使用例としてマニュアルから次の2つの例を抜粋し、ソースプログラムをそれぞれ次のファイルに登録している。

EXMAPLE/GENGO/PASCAL/S1 (マニュアルFGK01 p. 51 参照)

EXAMPLE/GENGO/PASCAL/S2 (マニュアルFGK01 p. 52 参照)

## 12. Reduce

Reduceは数式処理言語の一つである。本センターで使用できるものは東京大学より譲渡された第2版であり、SLISPの上で動作する。バッチ処理およびTSS処理で使用できる。TSS処理での処理方式はインタプリタである。センターニュースに解説記事<sup>2)</sup>が掲載されているのでそれを参照していただきたい。オリジナルの版には学習用のインタラクティブ・レッスンがあるが本センターでは使用できない。Reduceではオペレーターの実行にもmsecのCPU時間を要する場合があるので注意が必要である。

### 12.1 TSS処理

システム選択レベル(SYSTEM?)またはビルドモード(\*)で\$REDUをタイプインする。一度Reduceが起動されると、これが終了されるまでは他のサブシステムを使用できない。使用例のソースプログラムを下記のファイルに登録しているので、例えば代替名07でAFTに登録後、REDUCEを呼び出し、'IN 07;'をタイプインすると実行される。

EXAMPLE/GENGO/REDUCE/S1

### 12.2 バッチ処理

	1	8	16
1.	\$	JOB	000000000\$password,job-class
2.	\$	CPROC	REDUC
3.	\$	LIMITS	ttt,,,ppppp
4.			ソースプログラム
5.	\$	ENDJOB	

ソースプログラムをパーマネントファイルに登録している場合は、\$PRMFL文のファイルコードに05を使用する。

### 13. ShapeUp

ShapeUp は、日本電気の Prolog 言語であり、TSS 処理でのみ使用できる。機能的にはエジンバラ大学で開発された DEC-10 版 Prolog のサブセットであり、文字列処理の機能に SNOBOL の機能をほとんど含んでいる。処理方式はインタプリタである。

#### 13.1 使用法

ShapeUp は、使用する端末の型によって、処理系を起動するコマンドが異なる。システム選択レベル (SYSTEM?) に対し、ASCII 端末の場合は \$SHP1、JIS 端末の場合は、\$SHP3 をタイプインする。一度このサブシステムが起動されると、終了されるまで他のサブシステムを使用できない。

#### 13.2 エスケープシーケンス

JIS 端末では次の 4 文字が矢印の右側のエスケープシーケンスをもつ。

~ → & , | → \$ , { → 「 , } → 」.

#### 13.3 使用例

使用例をインタプリタの起動を含めて、次のコマンドファイルに登録している。これを実行するにはシステム選択レベル (SYSTEM?) またはビルドモードで、このファイルを入力ファイルとする CRUN コマンドをタイプインする。

ASCII 端末用    EXAMPLE/GENGO/SHAPEUP/C-A

JIS 端末用     EXAMPLE/GENGO/SHAPEUP/C-J

### 14. おわりに

本稿はプログラミング言語の利用の手引の初版として、その初歩を説明したものである。異種言語間の結合や入出力手法などに関する説明は本稿では省略しており、次の改版の機会にとりいれたいと考えている。本稿では、ACOS システム 1000 のバッチ処理の基本的 JCL および TSS 処理のコマンドに関する知識を前提としているため、やむをえず不親切な手引となっていることを、お許しいただきたい。実際に使用される場合は、マニュアルまたは仕様書を必ず参照していただきたい。これを機に、Fortran 以外の種々のプログラミング言語が使用されるようになれば、幸いである。

#### 参 考 文 献

日本電気(株)提供のプログラミング言語の参考文献は、センターニュース本号の巻末の日本電気(株)発行のマニュアルである。APL, LISP1.5, C, PASCAL では、文法説明書は発行されておらず、使用説明書のみとなっている。以下にはマニュアル以外のものを示す。特に \* 印のものは本センターの図書資料室に備えられていて、必要ならば借り出すことができる。

- 1) 辻野：計算機言語Cの役割，大阪大学大型計算機センターニュース，Vol. 14, No.1 (1984)。
- 2) 山本：Reduceの阪大センターにおける利用について，大阪大学大型計算機センターニュース，Vol. 12, No.4(1983)。
- 3) 杉山，中西：プログラミング言語XXPLの紹介，大阪大学大型計算機センターニュース，No.38 (1980)。
- \*4) LISP User's Manual，電子技術総合研究所(1979)。
- \*5) ShapeUp仕様書(V1)，日本電気C&Cシステム研究所(1983)。
- 6) 塩野：FORTRANとの比較によるPL/1入門(1)，大阪大学大型計算機センターニュース，No.36(1980)。
- 7) 塩野：FORTRANとの比較によるPL/1入門(2)，大阪大学大型計算機センターニュース，No.37(1980)。
- 8) 塩野：FORTRANとの比較によるPL/1入門(3)，大阪大学大型計算機センターニュース，No.38(1980)。
- 9) 塩野：FORTRANとの比較によるPL/1入門(4)，大阪大学大型計算機センターニュース，No.39(1980)。
- 10) 塩野：FORTRANとの比較によるPL/1入門(5)，大阪大学大型計算機センターニュース，No.40(1981)。
- 11) 塩野：FORTRANとの比較によるPL/1入門(6)，大阪大学大型計算機センターニュース，No.42(1981)。
- 12) 大中，後藤：FORTRAN77(V)概説(1)，大阪大学大型計算機センターニュース，Vol. 11, No.3(1981)。
- 13) 大中，後藤：FORTRAN77(V)概説(2)，大阪大学大型計算機センターニュース，Vol. 11, No.4(1982)。
- 14) 大中，後藤：FORTRAN77(V)概説(3)，大阪大学大型計算機センターニュース，Vol. 12, No.1(1982)。
- 15) 大中：FORTRAN77(V)のススメ，大阪大学大型計算機センターニュース，Vol. 12, No.3(1982)。
- 16) 後藤，大中：FORTRAN入出力からみた記録の長さ，大阪大学大型計算機センターニュース，Vol. 13, No.2(1983)。
- 17) FORTRAN77のVモードにおけるインライン展開について，大阪大学大型計算機センター速報，No.94(1983)。
- 18) ASLとMATHLIB-6について，大阪大学大型計算機センター速報，No.107(1983)。
- 19) 北本，他：漢字端末について，大阪大学大型計算機センターニュース，Vol. 13, No.3(1983)。
- 20) 後藤：FORTRAN77(V)における日本語文字データ処理，大阪大学大型計算機センターニュース，Vol. 13, No.4(1984)。
- 21) TSSの手引，大阪大学大型計算機センター(1984)。
- 22) SALS英文マニュアル，大阪大学大型計算機センター(1984)。
- 23) 泉谷，他：統合アレイプロセッサについて，大阪大学大型計算機センターニュース，Vol. 12, No.1(1982)。
- 24) 藤井：新しいオペレーティング・システムにおける仮想記憶機能，大阪大学大型計算機センターニュース，No.39(1980)。

(第4章の内容には，その後相当の変更がある。)