The University of Osaka
Institutional Knowledge Archive

Title	SX FORTRAN77概要(2) : 利用法を中心として
Author(s)	後藤, 米子; 大中, 幸三郎
Citation	大阪大学大型計算機センターニュース. 1986, 61, p. 37-53
Version Type	VoR
URL	https://hdl.handle.net/11094/65692
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

資　料

SX FORTRAN77 概要（2）

― 利用法を中心として ―

大阪大学大型計算機センター研究開発部
後藤米子・大中幸三郎

1. はじめに

本稿では、“SX FORTRAN77概要（1）”[3]（以下、概要（1）と略す）の続編として、SXでジョブを実行させるための簡易形の利用法について述べます。略語は概要（1）と同一のものを使用しています。

2.では利用者のファイルはすでにシステム内に登録済みであるという前提で、これをプログラムで使用する方法などを説明しています。**3.**ではバッチ処理における使用法を、**4.**ではTSS処理における使用法を、SX固有のジョブ制御言語（Job Control Language ; JCLと略記）とコマンドを含めて解説しています。**5.**ではコンパイラオプションの名称と機能の概要を、説明の都合上、主としてHFPのFORTRAN77[17,18]（HFP77と略す）との比較で示します。**6.**でデバック機能および言語仕様差の指摘ツールなどについて、**7.**では拡張領域の説明とその使用上の注意事項を述べます。

SXはHFPと同系統のシステムですから、HFPで実行可能であったプログラムはほとんど書きかえることなくそのままSXで実行可能です。ただし、APの性能を十分に引き出すにはプログラムのベクトル化率[1]を高めることが重要となりますから、プログラミング上の工夫が必要となります。これについては、センターニュース本号の別稿[4]を参照して下さい。利用者プログラムの性能向上のための各種支援ソフトウェアツールがありますが、これらについてはセンターニュースの次号に掲載される予定です。また、コンパイラオプションの既定値などの運用に関連した事項は1986年4月に速報[11]で広報される予定ですので、利用に際してはそれを併用してください。

2. SXジョブの実行環境

2.1 SXジョブの投入と取り出し

SXはACOS 1000のバックエンドプロセッサとして接続されており、HFPの場合と同様にすべてACOS 1000を経由して処理がおこなわれる。[1,2] 利用者によるSXジョブの投入と実行結果の取り出しに関する操作の手順は、従来のACOS 1000のジョブ処理の場合と同様であり、またHFPとも同様である。たとえば、SXジョブの投入を端末から会話型リモートバッチ処理でおこなうときにはCARDINサブシステムを使用し、その結果の取り出しにはJOUTサブシステムを使用す

るなどである。ただし、個々のJCLはACOS 1000のものが必ずしもそのままでは使用できず、またHFPとも異なるので注意が必要である（**3.**および**4.**参照）。

2.2 パーマネントファイル（データの場合）の保存と利用

SXジョブで使用するための利用者のパーマネントファイルは、ACOS 1000側の磁気ディスク上、SX側の磁気ディスク上、の双方に登録することができる。HFPではACOS 1000側にだけ登録することができた。それぞれのファイルは表**1**に示すように、ファイルの形式、接続の方法のすべてにわたって差があるので、注意しなければならない。ファイル管理はファイルが登録されているシステムごとにおこなわれるため、ACOS 1000側のファイルはオペレーティングシステムACOS-6のもとで、一方、SX側のファイルはオペレーティングシステムSXOSのもとで異なった管理を受ける。

表**1**　SXジョブで使用できるパーマネントファイル

	順編成ファイル		直接編成ファイル		接　　続
	書式付き	書式なし	書式付き	書式なし	
ACOS1000	可	不可	不可	不可	$ PRMFL文，SXRUNのFILE=
SX	可	可	可	可	OPEN文

(1) ACOS 1000側にパーマネントファイルが登録されている場合

この場合はファイルの形式が順編成書付きでなければならないこと、事前の接続だけが使用できOPEN文を使用できないこと、などを除いてAF 77Vとの差はなく、ジョブ処理の際には次のように扱われる。事前の接続のファイルfがあれば、ジョブの実行が開始されるまえにシステムによって自動的にSX側に対応したファイルf'が作成され、fの内容はコード変換などの操作をほどこされf'に転送される。SXジョブの実行にはf'が用いられ、終了後には自動的にf'は消去される。ただし、割当てタイプにWRITEが指定されている場合（例えば$ PRMFL文の第2パラメータがWである場合）には、ジョブの実行が終了し次第、消去のまえにf'の内容は上記と逆の操作をほどこされもとのfに転送（これを逆転送という）される。入力データとしてのみに使用する場合は、割当てタイプにREADを指定しておくことが望ましい。

(2) SX側にパーマネントファイルが登録されている場合

ファイルの形式はAF 77Vの場合と同じものが使用できるが、接続はOPEN文でおこなう。このときファイル記述子にはSXのものを指定する。OPEN文、CLOSE文、INQUIRE文の使用法にはAF 77Vの場合より制限があるので、概要(1)の**11.5**, **11.6**, **11.7**を参照していただきたい。なお、SX側にファイルを作成する方法については、別稿[5,6]に記述される予定である。

2.3 テンポラリファイルの利用

テンポラリファイルの利用は事前の接続およびTSS処理のOPEN文で可能であり、ファイルの形式はAF77Vと同一のものが使用できる。HFPの場合との差はない。

2.4 ソースプログラムファイルの利用とその形式

ソースプログラムはACOS 1000側のファイルに登録することができる。SX側のファイルに登録しこれをプログラムとして利用できるのは1986年秋頃になる予定である。したがって、以下ではACOS 1000側にソースプログラムファイルが登録されているときの扱われかたについて述べる。

システムによる取扱いは**2.2**の(1)に準じるが、割当てタイプの指定にかかわらず逆転送は行われない。SX77ではAF77Vの固定形式、行番号付き自由形式および行番号なし自由形式が処理できるので、次の差を除外すればソースプログラムの修正は不要である。HFP77との差はない。

① コンパイラオプションの名称(表**2**参照)が異なる。

② 1行の長さの最長桁数がSX77では行番号付き自由形式の場合に短くなる(表**2**参照)。

③ SX77では1行を72桁の固定長形式とみなす。AF77Vでは可変長形式とみなす場合もあった。

④ SX77では行番号の有無は自動判定される。AF77Vでは行番号の有無に応じてコンパイラオプションを指定しなければならなかった。

⑤ $ SELECTA文で行番号付きファイルを使用する場合は、行番号を除去する応答をしなければならない。

②の“1行の最長桁数”とは、1行に記述できる備考(例えば固定形式において73桁から80桁に記述するもの)以外の文字列の最大の長さのことである。③より、例えばアポストロフィで囲んだ一つの文字定数が複数行にわたる場合には、エラーとして検出される可能性がある。

⑤は、$ PRMFL文と$ SELECTA文のACOS1000本来の機能の差に起因している。すなわち、$ PRMFL文では指定されたファイルそのものが転送されるのに対し、$SELECTA文では、指定されたファイルからシステムの作業用のファイル(SYSINファイルとよばれる)が作

表2 ソースプログラムの形式

形式	1行の最長桁数		コンパイラオプション	
	SX77	AF77V	SX77	AF77V
固定形式	72	72	FIXED	FORM
行番号付き自由形式	72 (注1)	80 (注2)	FREE2 (注3)	LNO, NFORM
行番号なし自由形式	72	72	FREE2 (注3)	NLNO, NFORM

(注1) 行番号を含まない。行番号の直後の桁が第1桁とみなされる。
(注2) 行番号(1-8桁)を含む。
(注3) 行番号の有無は自動判定される。

成されこれが転送されることになるが、SX77ではこのファイルには行番号が許されないためである。

2.5 ライブラリの使用法

SX77で使用できる数値計算ライブラリは、日本電気提供のASL/SXとMATHLIB/SXである。前者には、高速化のためにアルゴリズムの改良とサブルーチンの追加がある。センターライブラリは使用できない。

バッチ処理における使用法はACOS1000およびHFPと同じであり、$ GO文のオプションに指定する。なお、結合するべきライブラリがSX77-AP用であるかSX77-CP用であるかは、ジョブクラスから自動判定されるので利用者は指定しなくてよい。TSS処理では**4.**に示すように変更があり、**表3**に示すライブラリ名に対応したファイルコードをSXRUNコマンド(**4.2**参照)のオプションに指定する。

表3 ライブラリ使用のためのオプション

ライブラリ名	$GO文	LIB=オプション
ASL/SX	ASL (SX77-AP用) ASL (SX77-CP用)	AA(SX77-AP用) AC(SX77-CP用)
MATHLIB/SX	MLIB(SX77-AP用) MLIB(SX77-CP用)	MA(SX77-AP用) MC(SX77-CP用)

3. バッチ処理における使用法

3.1 ジョブの構成

SXにおいて簡易形でバッチ処理を行う場合には、基本的にはACOS1000のJCLに$ SX文を追加すればよい。ジョブの構成におけるACOS1000のJCLとの相違点は、次のとおりである。

- $ JOB文

  ジョブクラスには、SX77-APを使用する場合はUまたはVを指定し、SX77-CPを使用する場合はSまたはTを指定する。同一ジョブにおいてACOS1000上で処理するアクティビティと、SX上のアクティビティが混在してはならない。レポート指定Rは指定しても意味をもたない。

- $ LIMITS文

  使用不可。**3.2**の$ SX文のオプションを参照のこと。

- $ FRT77文

  コンパイラオプションが異なる(**5.**参照)。

• $ GO文

一つのジョブでは$ GO文は一つしか使用できない。オプション欄ではライブラリの指定のみを行う。

• $ PRMFL文

ACOS 1000側のパーマネントファイルを使用するときに指定する。ファイル記述子はACOS 1000と同一のものを指定する。直接探査のパーマネントファイルを使用できないため、第3パラメータにRを指定してはならない。ファイル転送に関連していくつかのパラメータが追加されている。

• $ SYSOUT文

他のJCLではファイルコードは01～99まで指定できるのに対し、$ SYSOUT文では01～63に制限されている。次回のバージョンアップで、01～99まで指定可能となる予定である。

図1にCARDINサブシステムを用いて、会話型リモートバッチ処理をおこなう場合のJCLの例を示す。センターのカードリーダからジョブを投入する場合には、図1のJCLの最初に$ SNUMBカード、最後に＊＊＊EOFカードをつければよい。

1カラム	8カラム	16カラム
$	JOB	利用者番号；支払コード$パスワード，ジョブクラス等
$	SX	オプション
$	FRT77	オプション
	ソースプログラム	
$	GO	オプション
	データ	
$	ENDJOB	

図1　会話型リモートバッチ処理のJCLの例

ファイルの転送および逆転送の制約から、SXで実行可能な相対形式プログラムおよび実行形式プログラムの保存はできない。

3.2　$ SX文

$ SX文はジョブをSX上で処理することを宣言し、その実行時の演算時間と出力記録数の上限を指定するものである。$ SX文の位置は$ JOB文の直後であり、省略することはできない。$ SX文の形式を次に示す。$ SX文は$ ETC文によって継続することができるが、後述のように$ SX文のみに指定でき、$ ETC文には指定できないオプションがあるので注意しなければならない。

1カラム	8カラム	16カラム
$	SX	オプション

オプションには、下記のものをコンマで区切って、任意の順序で指定する。下記以外のオプションを指定してはならない。

- TYPE=*ty*

  利用形態を*ty*で指定する。簡易形（本稿で述べるような利用法）で使用する場合にはA6を指定し、基本形で使用する場合にはSXを指定する。既定値はA6である。このオプションは$ SX文のみに指定でき、$ ETC文には指定できない。

- CPTIME=*t*

  SXにおける実行時の演算時間の制限値*t*を秒単位で指定する。ACOS 1000 では原則として100分の1時間単位で指定していた。既定値は60秒である。

- OUTLIM=*p*

  SXにおける実行時のSYSOUTファイル出力記録数の制限値*p*を指定する。既定値は5000記録である。

- LIST=*str*

  SXのレポートの出力制御を*str*で指定する。実行レポートを出力する場合には*str*にSOURCEを指定し、出力しない場合には、NOを指定する。既定値はSOURCEである。

- STAGE=*st*

  プログラムの実行が異常終了したとき、WRITEパーミッションで指定されたファイルの内容を（ただし、内容が変更されたところまで）、ACOS－6側に転送するか否かを*st*で指定する。転送する場合にはYESを指定し、転送しない場合にはNOを指定する。既定値はYESである。

4. TSS処理における使用法

4.1 処理の流れ

SXのTSS処理ではSX77-CPのみが使用でき、ACOS1000のTSSのFRT77サブシステムのもとで、SXRUN（またはSXRUNH）コマンド（**4.2**参照）を入力することによって実行が開始される。すなわち、FRT77サブシステムのもとでRUN（またはRUNH）コマンドを入力すればACOS 1000で実行が開始されるのと同様である。ソースプログラムファイルは、バッチ処理の場合と同様にACOS 1000側に登録されているものを使用する（**2.4**参照）。カレントファイルから直接に実行できないので、編集処理後は一度パーマネントファイルに登録してから、SXRUNコマンドを投入しなければならない。

処理が終了すると、従来と同様に＊が表示される。この時点ではACOS 1000のTSS 処理をおこなうことができるし、再びSXRUNコマンドでSXのTSS処理を開始することもできる。

SXRUNHコマンドはACOS 1000のRUNHコマンドに相当し、日付、時刻および固有の進行メッセージを表示するもので、その他についてはSXRUNコマンドと同様である。以下の記述ではSXRUNコマンドについて述べるが、SXRUNHコマンドについても適用できる。

4.2 SXRUNコマンド

SXRUNコマンドは、SXでのTSS処理を起動するための実行コマンドであり、ACOS 1000で実行するためのRUNコマンドに相当する。SXRUNコマンドの形式は、オプションのキーワードの文字列などに一部変更のあることを除けば、RUNコマンドと同様である。

SXRUN　*fs* ：*olist*

fs　ソースプログラムのファイル記述子を指定する。カレントファイルからの実行はできないので、このパラメータは省略できない。

olist　コンパイラおよびリンカに対するオプションを指定する。区切り記号はRUNコマンドと同様、空白である。

以下にTSS処理固有の主なオプションについて述べる。コンパイラオプションについては**5.**を参照していただきたい。

(1) 変更されたオプション

- CPTIME＝*t*

  実行時の演算時間の制限値 *t* を秒単位で指定する。RUNコマンドのTIMEオプションと同一の機能である。既定値は60秒である。

- LIB＝*fc* またはL＝*fc*

  実行時に使用するライブラリ（**2.5**参照）をそのファイルコード *fc* で指定する。複数のライブラリを使用する場合は、ファイルコードの並びをコンマで区切る。

- FILE＝*filedescrp*(*fc*), *olist* または F＝*filedescrp*(*fc*), *olist*

  ファイル転送またはSX側のファイル作成などに関連して、いくつかのパラメータが追加されている。ただし、*filedescrp* はファイル記述子、*fc* はファイルコードを示す。*olist* にはファイル属性に関するオプションを指定する。

(2) 変更のないオプション

- TERMINAL＝*fc* またはTERM＝*fc*

4.3 注 意 事 項

- 事前の接続をする場合はSXRUNコマンドのFILEオプションで指定しなければならない。

FILEオプションで指定されたファイルの扱いは**2.2**の(1)および**2.3**で述べたとおりである。

ACOS 1000 ではファイルコードがAFTに登録されている場合には事前の接続とみなされていたが、SXではこのようなことはなくなる。

- オプションDBLINEを指定するときには、＊＃行の文字列＊＃の代わりに文字列C＃を使用しなければならない。AF77Vでは＊＃行はオプションの影響を受けなかった。
- 入力促進記号がAF77Vの場合とは異なる。

5. コンパイラオプション

本節では、説明の都合上、主としてHFP77との比較について述べる。本節と次節では“HFP77と同一”という表現は利用者から見た場合にプログラムにおよぼす効果などが同一であることを意味し、内部処理の詳細までが同一であることを意味しない。SXのFORTRAN77のコンパイラには、SX77-AP用とSX77-CP用の2種類がある（概要(1) **2.1**, **2.2** 参照）。APとCPの機能の差から、オプションにはSX77-APとSX77-CPの双方で共通に使用できるものと、一方のみで使用できるものとがあるので注意が必要である。SX77-APではベクトル化関連のものが指定でき、SX77-CPでは、HFP77からIAP関連を除外したものがおおむね使用できる。両者ともにAF77V[14]と比較するとかなりの部分が異なっているので、ACOS 1000から移行する場合には注意しなければならない。

オプションの指定箇所は、バッチ処理では $ FRT77 文のオプション欄であり、TSS処理ではSXRUNコマンドのオプション欄である。バッチ処理では、SX77-APを使用するかSX77-CPを使用するかの区別はジョブクラスから自動判定されるので、利用者は指定しなくてよい。TSS処理ではSX77-CPのみが使用できるので本節の内容の一部に限定されることに注意されたい。

以下に、コンパイラオプションの主なものを示す。とくに断りのないものは、SX77-APとSX77-CPの双方に共通である。オプションの名称のうち、機能を無効とするためのものは“NO”を先頭に付加するように統一されている。ACOS1000では主に“N”を用いていた。デバック用のオプションと併用する場合の注意事項については、**6.1** を参照していただきたい。*plist* はオプションの付随パラメータを示し、コンマで区切って任意の順序で指定する。運用の既定値は1986年2月現在で未定であり、別途、速報[11]に掲載する予定である。

5.1 HFP77と同一のオプション

以下のオプションは、下記に示す理由によりAF77Vと異なるものである。

(1) 対応する機能がAF77Vにあるがオプション名が変更になるもの

括弧内はこれと機能的にほぼ対応するAF77Vのものである。

- SOURCE(LSTIN), NOSOURCE(NLSTIN)
- FIXED(FORM), FREE2(NFORM)

- DOLLAR(DLR), NODOLLAR(NDLR)
- EJECT(EJECT), NOEJECT(NEJECT)
- ALC(ALC), NOALC(NALC)注)
- BYNAME(BYNAME), NOBYNAME(NOBYNAME)
- DBLINE(DBGLN), NODGLINE(なし)

注) AF77VではNALCオプションは使用できない。

(2) 対応するオプションがAF77Vにないもの

- COLLATE=EBCDIC, COLLATE=JIS
- SCHAR, LCHAR (SX77-CPのみ)
- LCASE, NOLCASE
- TEST, NOTEST (SX77-CPのみ)
- MAIN=*name*, BLKDATA=*name*
- OBSERVE, WARNING, SERIOUS, FATAL
- CONSTEXT=ALL, CONSTEXT=NOEXP, CONSTEXT=NONE
- FLAG, TITLE, AUTOTRUNC(SX77-CPのみ), MRGMSG, ELN, ILN

(3) 対応するオプションの仕様がAF77Vから変更されたもの

- AUTODBL=値
- OPT=値 (SX77-CPのみ)

(4) AF77Vでは指定していたがSX77では標準機能であるため使用できないもの

- ALIGN, NALIGN
- ASTER, NASTER
- ARGCHK
- ASMDARY

(5) SX77では対応する機能がないために使用できないもの

- ROUND, NROUND
- MAXLIN

5.2 HFP77から仕様変更されたオプション

- INLINE

SX77-APでは4倍精度実数型の除算はインライン展開の対象とならない。SX77-CPではHFP77と同様にインライン展開される。

- MASK=(*plist*)

SX77-CPではHFP77と同一のパラメータが指定できその機能も同一であるが、SX77-APでは機能が拡張されたためにパラメータに指定できるものが変更されている。

5.3 HFP77から削除されたもの

(1) SX77-AP, SX77-CP双方ともに削除されたもの

- IAP, NOIAP

統合アレイプロセッサ（Integrated Array Processor ；IAP）がなく、ベクトル処理はAPのベクトルユニットでおこなわれるため、VECTORオプション（**5.4**(2)参照）などを使用する。

- ARGSKIP, NOARGSKIP

  将来は使用できるようになる予定であるがその時期は未定である。

(2) SX77－AP のみが削除されたもの

- REORDER, NOREORDER

  SX77－CPではHFP77と同一であるが、SX77－APではCHECKオプション（**6.1.1**参照）指定時を除いてつねにREORDERとなっており、オプションとしては存在しない。

5.4 HFP77から新規に追加されたオプション

(1) SX77－AP，SX77－CP共通のもの

- XAREA＝nまたはXAREA＝mM，NOXAREA（ただし、SX77－APでは$1 \leqq n \leqq 4096$，$1 \leqq m \leqq 4$，SX77－CPでは $64 \leqq n \leqq 4096$，$1 \leqq m \leqq 4$）

  XAREAでは、共通ブロックまたは局所配列の大きさがnKバイト（またはmMバイト）を越えたとき拡張領域（**7.**参照）に割り当てることを指定し、NOXAREAでは、拡張領域を使用しないことを指定する。SX77－APでは大きさの値の変更のみができ、NOXAREAの指定はできない。

- DBG, NODBG

  6.1.2参照。

(2) SX77－AP固有のもの

各オプションの付随パラメータにはHFP77とそのキーワードおよび機能が同一のものもあり、これらについては説明を省略する。

- VECTOR＝（*plist*），NOVECTOR

  自動ベクトル化を適用する（VECTOR）か、適用しない（NOVECTOR）かを指定する。*plist*には次のものを指定する。

  - FULLMSG, INFOMSG, NOMSG

    HFP77と同一である。

  - VERRCHK, NOVERRCHK

    ベクトル化された組込み関数の引数の値のチェックをする（VERRCHK）か、しない（NOERRCHK）かを指定する。HFP77にはこの機能はなかった。

  - DIVLOOP, NODIVLOOP

それぞれHFP77のEXTENDとBASICにほぼ対応し、その機能もほぼHFP77と同様である。

- LOOPCNT＝l

  lの値の既定値も含めてHFP77と同様である。

- VWORKSZ＝nまたはVWORKSZ＝mM　ただし、$1 \leqq n \leqq 4096$,　$1 \leqq m \leqq 4$.

  HFP77のVECSEGSZに対応し、その機能はHFP77と同一であり作業用ベクトル領域の最大長を指定する。既定値は1Mである。nを指定した場合はnKバイト、mMを指定した場合はmMバイトとなる。HFP77の場合と比較して、この作業領域が必要となるケースは少い。

  HFP77の場合はnの形のみを指定でき、その既定値は64であった。HFP77の場合と同様に、この値が不足するとベクトル化が可能の場合でもベクトル化されないので注意しなければならない。

- VDIR, NOVDIR

ソースプログラム中のベクトル化指示行（本稿**5.5**および概要(1)**15.3**参照）を有効とする（VDIR）か、注釈行とみなす（NOVDIR）かを指定する。これら2つのオプションの効果は＊VDIRの付随パラメータも含めてコンパイラオプションのVECTORおよびNOVECTORに優先する。

ベクトル化指示行の先頭の文字は＊であるが、DBLINEおよびNODBLINEオプションの影響を受けない。ただし、先頭の5文字が＊VDIRであるような注釈行およびデバッグ行を書いてはならない。

- XOPT＝（*plist*）

SX77－APでは、AF77VおよびHFP77と異なり、最適化のレベルを利用者が指定できずつねに最大限の最適化が適用される。ただし、CHECKオプション指定時には、SX77－CPのOPT＝1のレベルとなる。下記の付随パラメータは、最適化の副作用がおきる可能性を回避するために指定する。

  - DIV, NODIV

    HFP77と同一である。

  - MOVE, NOMOVE

    HFP77と同一である。

  - CHG, NOCHG

    式の評価順序の変更を行う（CHG）か、行わない（NOCHG）かを指定する。HFP77にはこのオプションは存在しなかった。

- INLOG4, NOINLOG4

1バイト論理型および2バイト整数型を、それぞれ、4バイトのものとみなす(INLOG4)か、型宣言に従う(NOINLOG4)かを指定する。これらの型はベクトル化の対象外であるため、ソースプログラムの変更なしに型を変更するためのものである。HFP77にはこのオプションは存在しなかった。

- MASK=(*plist*), またはSETMASK=(*plist*), NOSETMASK

演算例外の検出をおこなうか抑止するかの制御を下記の *plist* で指定する。MASKオプションはプログラムの実行開始の時点で(すなわち主プログラムに対して)*plist*を指定するもので、SETMASKオプションはそれを副プログラム内において変更するときに使用する。実行時の効率は実行可能プログラム全体が同一の制御に従うほうがよい。

- ZDIV, NOZDIV　• FLOVF, NOFLOVF　• FLUNF, NOFLUNF
- FXOVF, NOFXOVF

抑止するとエラーメッセージは出力されず、例外に対応したインディケータもセットされないので、組込みサブルーチンによる例外の検出は不可能となる。ASL/SX, MATHLIB/SXは、NOSETMASKオプションでコンパイルされている。HFP77およびSX77−CPでは、浮動小数点演算のオーバーフロー例外を抑止することはできず、固定小数点演算の零除算例外とオーバーフロー例外とを分離して検出できなかった。

5.5 ベクトル化指示行

ベクトル化指示行を使用する主な目的は、コンパイラの自動ベクトル化の補助として、コンパイラによるソースプログラムの解析では得られない情報を利用者が与えることでベクトル化の割合を向上させるためのものである。これとは逆にベクトル化を抑制するときにも使用可能であるが、このような場合は稀であるので本稿では前者の場合のみについて述べる。以下ではコンパイラオプションのVECTORが有効となっていることを前提としている。

ベクトル化指示行の形式は、次のとおりである。

1カラム
*VDIR *plist*

付随パラメータ *plist* には次頁のものをコンマで区切って任意の順序で指定する。(1)はコンパイラオプションのVECTORの付随パラメータと同じであり、それと異なる値を指定する必要がなければ省略してもよい。(2),(3),(4)は利用者がこの指示行のみで与えられる情報である。指定箇所とその有効範囲はその与える情報によって異なる。(1)および(2)ではDOループの直前に指定し、直後のDOループに対して有効となる。(3)および(4)では指定された手続きが引用される以前であれば、プログラム単位内の任意の箇所に指定でき、プログラム単位全体に有効となる。

(1) コンパイラオプションVECTORと同一のもの。

FULLMSG, INFOMSG, NOMSG, LOOPCNT, VERRCHK, NOVERRCHK, VWORKSZ, DIVLOOP, NODIVLOOP.

これらはコンパイラオプションで指定されたものに優先する。

(2) NODEP（*alist*）　*alist*はコンマで区切った配列名の並び。

*alist*中に現れた配列名について、データの定義・引用の関係が、ベクトル化可能な性質をもつことをコンパイラに指示する。NODEPだけを指定すると、この性質をもつかどうかの判断がつかない場合には、すべてベクトル化可能であることを意味する。

(3) NORMAL（*eplist*）　*eplist*はコンマで区切った外部手続き名の並び（省略不可）。

外部手続きが、最適化処理またはループ分割によるベクトル化が可能な性質をもつことを指示する。

(4) ELEMENTAL（*flist*）　*flist*はコンマで区切った外部関数名の並び（省略不可）。

外部関数が、最適化処理またはベクトル化が可能な性質をもつことを指示する。この性質をもつための条件については省略する。

6. その他

6.1 デバッグのためのコンパイラオプションと宣言文

SXシステムに用意されているデバッグのためのツールには、次のものがある。

(1) 添字の値および文字位置式の値の検査（**6.1.1**参照）

(2) デバッグのための宣言文（**6.1.2**参照）

(3) デバッグサポートプログラム（**6.1.3**参照）

(4) デバッグ行（**6.1.4**参照）

(1),(2)はSX77-APおよびSX77-CPで使用可能であるが、その実行環境などに差がある。(3)はSX77-CPのみで使用可能できる。(4)はSX77-APおよびSX77-CPで使用可能である。機能および注意事項は次のとおりである。

6.1.1 CHECKオプション（SX77-AP, SX77-CP共通）

AF77VのSUBCHKオプションに相当する。本オプションと、最適化機能およびその付随機能とは次の関係がある。

- SX77-APでは、最適化機能は自動的にその大部分が抑止され、SX77-CPでのOPT＝1と同程度となる。XOPTのパラメータを指定しても無視され、VECTORまたはVDIRオプションが指定されると、それぞれ、NOVECTOR, NOVDIRが指定されたものとみなされる。

- SX77−CPでは、最適化オプション（すなわちOPT＝0, 1, 2, 3）を同時に指定した場合は、OPT＝0を指定したものとみなされる（HFP77と同一）。
- SX77−AP，SX77−CPともにINLINEオプションを指定しても無視され、NOINLINEオプションを指定したものとみなされる。

6.1.2 DBG, NODBGオプションとデバッグのための宣言文（SX77-AP, SX77-CP共通）

SX77で新規に追加された機能であり、下記の宣言文（概要（1）7.4参照）を有効とする（DBG）か、無効とする（NODBG）かを指定する。CHECKオプションが同時に指定された場合は、DBGオプションは無視され、NODBGオプションが指定されたものとみなされる。

① SUBCHECK文

SUBCHECK文は、CHECKオプションの機能のうちの添字の値の検査の適用範囲を特定の配列名またはプログラム単位の一部に限定したり、特定のプログラム単位に限定するときに使用する。

② SUBSTRCHK文

SUBSTRCHK文は、CHECKオプションの機能のうち、文字位置式の値の検査の適用範囲を特定のプログラム単位に限定したいときに使用する。

本オプションと最適化機能およびその付随機能との関係は次のとおりである。

- SX77−APでは検査の対象となる配列を含むDOループはベクトル化の対象外となるが、これら以外は通常どおりベクトル化が適用される。文字位置式を含まないプログラムにSUBSTRCHKを指定しても何の影響もない。
- SX77−CPでは上記のCHECKおよびTESTオプションの場合と同様である。

6.1.3 TESTオプション（SX77-CPのみ）

AF77VのFDS（FORTRAN Debugging System）に相当するデバッグサポートプログラムを使用するときに指定する。本オプションと最適化機能およびその付随機能とは上記のCHECKオプションと同様の関係がある。

6.1.4 DBLINEオプション（SX77-AP, SX77-CP共通）

デバッグ行を有効にするときにDBLINEを指定し、注釈行とするときにNODBLINEを指定する。DBLINEおよびNODBLINEオプションの有無にかかわらず＊VDIRで始まるデバッグ行を書いてはならない。

6.2. 仕様差指摘ツール FLINTER

SX77においても、HFP77における場合と同様に、言語仕様の差を指摘するためのソフトウェアツールが用意されている。指摘項目はAF77VとSX77との差であり、原則としてSX77−APとSX77−CPの共通仕様の範囲に限られる。AF77Vで書かれたソースプログラムを入力とし、この

中のSX77との言語仕様差を自動的に診断し、その情報をリスト上に出力する。指摘メッセージは、該当するソースプログラムの文の直後に表示される。HFP77との仕様差を指摘するツールはない。

使用法は下記に示すとおりであり、バッチ処理では$FLINT文、TSS処理ではFLINT コマンドを使用する。オプションについては、省略する。

- バッチ処理

1カラム	8カラム	16カラム
$	JOB	利用者番号；支払コード$パスワード，ジョブクラス等
$	FLINT	オプション
	ソースプログラム	
$	ENDJOB	

- TSS処理

SYSTEM? FLINT ソースプログラムファイル：オプション注)

注）画面端末では、オプション欄にSEDITの指定が必要である。

7. 拡張領域

利用者から見た記憶領域の論理的な取扱いについては、SX77とAF77Vの間で差異がある。すなわち、SX77では一つの実行可能プログラムの記憶領域は以下のように類別されている。

基本領域	小セグメント	最大64KBかつ255個以下
	大セグメント	最大4MBかつ4個以下
拡張領域	拡張セグメント	

SX77ではコンパイラが出力するコンパイルユニットは、コード部分とデータ部分に区別される。コード部分は、一つの大きさが64KB以下となるように区切って作成され、基本領域に割付けられる。一方、データ部分は適当に区切れないものがある。すなわち、以下に示す三つの場合は、記憶列として同一のセグメント内に連続に割当てられなければならず、複数のセグメントに区切ることができない。

配　列

共通ブロック

EQUIVALENCE文によって結合された変数や配列など

データ部分の中で上記の記憶列の大きさがXAREAオプションで与えられた値（既定値4MB）を越える場合には拡張領域に割付けられ、それ以外の場合は基本領域に割付けられる。

コンパイラによって作成されたコンパイルユニットは、リンカによって実行可能プログラムとして結合される。このとき、コード部分は小セグメントに割当てられる。データ部分の中で基本領域

に割付けられたものはその大きさに応じて大、小セグメントに割当てられ、拡張領域に割付けられたものは拡張セグメントに割当てられる。また、同じ属性をもつセグメント群はリンカの自動バインド機能によって、セグメントの個数の減少とその大きさの上限までの有効利用が図られる。

利用者が注意しなければならないことは、拡張領域に割付けられたデータに対して初期値を与えられないこと[3]以外には、大セグメントの個数の制限がきびしいことである。すなわち、XAREAオプションの既定値が4MBであることから、上記の割当て規則においては、他に大きな記憶列がない場合には次の三つの例は大セグメントに割当てることができる。

4MBの配列4個，2MBの配列8個，1MBの配列16個

しかし、3MBの配列が5個の場合には大セグメントの個数が不足する。このような場合にはXAREAオプションを3MBより小さく指定する必要がある。ただし、初期値を与えている場合には、ソースプログラムの変更が必要となる。小セグメントの個数が不足する場合はきわめて稀であるから、その回避策については省略する。

8. おわりに

本稿ではSXを利用する上での注意事項も含めて、初心者のかたにも理解していただくことを狙いとして述べてきました。HFPサブシステムの運用では、利用者のパーマネントファイルをHFP側に登録できなかったために、ジョブの性格によってはHFPを使用できない場合がありましたが、SXの運用ではこのような制約がほぼ全面的に解除されるために、HFPと同様のバックエンドプロセッサであるにもかかわらず、HFPの場合と比較して利用しやすい運用となっています。SXに関連した他の広報記事[1~6,11]とともに本稿がお役に立てば幸いです。

資料の提供などをいただいた日本電気株式会社基本ソフトウェア開発本部片山博主任ならびに関西日本電気ソフトウェア株式会社三原敏敬氏、河原久美子氏をはじめとする諸氏に感謝します。

参　考　文　献

SX関連

1）渡辺，近藤，端山，大中，藤井：スーパーコンピュータSX-1の概要（1），大阪大学大型計算機センター・ニュース，Vol. 15, No.4 (1986).

2）藤井：スーパーコンピュータSX-1の概要（2），大阪大学大型計算機センター・ニュース，Vol. 15, No.4 (1986).

3）大中，後藤：SX FORTRAN77概要（1），大阪大学大型計算機センター・ニュース，Vol. 16, No.1 (1986).

4）片山，河原，大中：FORTRAN77/SXにおける高速化技法，大阪大学大型計算機センター・ニュース，Vol. 16, No.1 (1986).

5）馬野：スーパーコンピュータSX-1のタイム・シェアリング・システムATSS-AFの使い方（その1），大阪大学大型計算機センター・ニュース，Vol. 16, No.1 (1986).

6) 多喜：ACOS-1000とSX-1のファイル転送について，大阪大学大型計算機センター・ニュース，Vol. 16, No.1 (1986).

7) GGB11-1 FORTRAN77言語説明書，日本電気(1986).

8) GGB12-2 FORTRAN77, 77/SXプログラミング手引書，日本電気(1986).

9) GDA12-2 SXCP解説書，日本電気(1986).

10) GJF11-1 MSF-6 利用説明書，日本電気(1986).

11) 大阪大学大型計算機センター速報，No.134 (1986).（予定）

ACOS 1000関連

12) 大中，後藤：FORTRAN77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) AGB01-4 FORTRAN77言語説明書，日本電気(1985).

16) FGB27-1 FORTRAN77プログラミング手引書，日本電気(1985).

HFP関連

17) 大中，後藤：HFP FORTRAN77概要(1)，大阪大学大型計算機センター・ニュース，Vol. 14, No.3 (1984).

18) 後藤，大中：HFP FORTRAN77概要(2)，大阪大学大型計算機センター・ニュース，Vol. 14, No.3 (1984).

19) FQA01-1 HFPサブシステム・利用手引書(標準形)，日本電気(1984).