

SDOS—43(II)アセンブラプログラムにおける 入出力処理

吉 村 卓

Input/output Program in SDOS—43(II) Assembler

Takashi YOSHIMURA
(昭和54年10月25日受理)

We consider some I/O techniques in assembler program, especially a method of using the executer of FORTRAN object program under the disc operating system SDOS—43(II) of the OKITAC—4300. In order to explain these techniques, we use some examples in section 2 and 3.

1. 緒 言

1971年にIntel社よりマイクロプロセッサ4004が発売されて以来、半導体技術の急速な発展により、マイクロプロセッサの品質の向上、小型化、軽量比が実現され、低価格のマイクロプロセッサの普及とあいまって広範囲にわたる応用製品が開発され続けている。情報処理システムの端末、計測装置や分析機器と連動したデータ処理システム、プロセス制御或いは自動製図機等の数値制御への応用など、従来コントローラとしてはミニコンピュータが使われていた分野にマイクロコンピュータが代って用いられるようになり、コンパクトかつ高性能のシステムが構成され、價格的にも充分採算のとれるものが供給されるようになった。加えて、キャッシュレジスタや自動販売機など商業的応用や、燃費節約のための点火時期の制御を始めとする種々の自動車制御への応用、或いは画像表示端末などデータ通信への応用等々、幅広い新応用分野が開発されつつある。⁽¹⁾

ところで、マイクロコンピュータのプログラミング言語としては、ミニコン以上の一般コンピュータと同様アセンブラはもちろん、PL/I系、FORTRAN、COBOL、RPG、BASIC等の高級言語も考えられているが、現時点ではアセンブラ言語が最も広く使われており、高級言語としてはBASICが普及しつつある状態である。しかし、ソフトウェア開発の比重が今後一層大きくなるであろうから、いずれ近い将来FORTRANなども広く使われることになるだろう。

しかし、コントローラの応用製品にあっては、制御プログラムは機器メーカーによって作成され、メモ

りに収容してユーザに提供されるので、たとえ高級言語が普及したとしても、マイクロコンピュータのメモリ容量を節約するためにアセンブラが多く用いられることには変りがないであろう。

ところで高専におけるプログラム言語教育の現状はどうであろうか。詳しい統計資料はないが、「高等専門学校における情報処理教育の実態調査報告書」⁽²⁾によれば、使用教科書一覧表から推して、電気工学科を除けば多くの学科ではFORTRANのみというところが大半なのではなかろうか。アセンブラ言語教育の重要性を再認識すべき時期にきているものと思われる。

アセンブラ教育が敬遠される理由の第1としては、アセンブラ言語が機種毎に全く異なるという点あげられる。しかし、教育の目的がプログラミング方法論の指導にあるとすれば、言語の違いは第2義的な問題と考えてよいであろう。

本校には現在HITAC—8250とOKITAC—4300の2機種が設置されているが、バッチ処理による実習を前提にアセンブラによるプログラミングを初学者に指導するとすれば、機械命令の種類が少ないミニコンシステムOKITAC—4300を用いて行なうのが学び易いであろう。しかるに、このシステムのオペレーティングシステムSDOS—43(II)アセンブラには入出力に関するシステムサブルーチンが用意されていないので、全くの初心者にとって入出力に関するプログラム作成を非常に困難なものにしている。

一方、SDOS—43(II)の管理下で機能するFORTRANに関しては、オブジェクト実行時の補助ブロック(主として入出力関係のルーチンから成る)があるのだから、これを同じくSDOS—43(II)の下でのアセン

ブラプログラムに利用できない筈はない。また、プログラムの生産性の面から考えても折角現にあるソフトウェア資源の有効利用を図ることは初学者に対してのみならず大切なことであろう。

次節以下にいくつかのプログラム例をあげて上記難題の解決策を考察することにしよう。

2. FORTRANサブプログラムによる方法

先づ第1に思いつく方法は入出力に関する部分をFORTRANのサブルーチン副プログラムで構成することである。プログラム例1はユークリッドのアルゴリズムによる2つの正整数A, Bの最大公約数Cを求めるプログラムである。主プログラムと副プログラムの間におけるデータの受け渡しはCOMMONによっている。

```

00001      SAP
00002      SUB          EUCLID
00003      COMMON     A, B, C
00004      TOP
00005      NO
00006      CAL          INPUT
00007      L           A
00008      T           N1
00009      L           B
00010      T           N2
00011      MOD
00012      SRD        15
00013      SRD         1
00014      D          N2
00015      XAR
00016      JZ          (A), GCD
00017      T           N3
00018      L           N2
00019      T           N1
00020      L           N3
00021      T           N2
00022      J          MOD
00023      L           N2
00024      CAL          OUTPUT
00025      J          I, TOP
00026      N1
00027      N2
00028      N3
00029      END

```

```

0001      SUBROUTINE INPUT
0002      INTEGER A, B, C
0003      COMMON A, B, C
0004      READ(5, 500) A, B
0005      500 FORMAT(2I5)
0006      RETURN
0007      END

0001      SUBROUTINE OUTPUT
0002      INTEGER A, B, C
0003      COMMON A, B, C
0004      WRITE(6, 600) A, B, C
0005      600 FORMAT(1H, 1H(, 15, 1H, , 15, 2H)=, 15)
0006      RETURN
0007      END

*EXECUTE, EUCLID
MEMORY, SIZE, 0754

( 782, 16)= 2
*END

*EOF

```

図1. プログラム例1

プログラム例2はニュートン法による正数Aの平方根Yを求めるプログラムである。この例ではプログラム間のデータ受け渡しは引数を用いて行なっている。なお、このプログラムではシステムライブラリの組込み関数ABSを用いている。システムの組込み関数や基本外部関数の利用についての詳細はOKI-TAC-4300 SDOS-43(II)の解説書を参照されたい。

```

00001      SAP
00002      SUB          SQRROOT
00003      TOP
00004      FL          A
00005      FA          C1
00006      FM          C2
00007      FT          X
00008      LOOP
00009      FL          A
00010      FD          X
00011      FA          X
00012      FM          C2
00013      FT          Y
00014      FS          X
00015      CAL          ABS
00016      FS          EPS
00017      JNZ          (S), EXT
00018      FL          Y
00019      FT          X
00020      EXT
00021      CAL          OUTPUT
00022      DEF          A
00023      DEF          Y
00024      J          I, TOP
00025      C1
00026      C2
00027      EPS
00028      A
00029      X
00030      Y
00031      DC          1.0
00032      DC          0.5
00033      DC          1.0E-3
00034      DC          78218.0
00035      BRV         2
00036      BRV         2
00037      END

0001      SUBROUTINE OUTPUT(A, Y)
0002      WRITE(6, 600) A, Y
0003      600 FORMAT(1H, 5HROOT(, F10.1, 2H)=, E18.8)
0004      RETURN
0005      END

*EXECUTE, SQRROOT
MEMORY, SIZE, 0968

ROOT( 78218.0)= 0.27967480E+03
*END

*EOF

```

図2. プログラム例2

プログラム例3は(l, m)行列Aと(m, n)行列Bの積Cを求めるプログラムである。配列要素を引用するには、先頭要素が格納されているメモリアドレスからの偏位値をBレジスタ修飾等の方法で指定しなければならないが、COMMON要素に対しては間接指定やBレジスタ指定等のアドレス修飾ができないので、配列をCOMMONによって引渡すことはできない。引数によるプログラム間のデータ授受の方法をとらざるを得ない。ところで、引数が配列の場合、パラメータアドレスは実引数の値が入っているアドレスではなく、アレイサブテーブルへのリンクアドレスである。アセンブラプログラムではアレイサブテーブルから配列要素の実際のアドレスを求めて処理することになる。配列要素のアドレスを求めるルーチンとしてシステムライブラリの中にE\$-

SDOS-43(II)アセンブラプログラムにおける入出力処理

ACというルーチンがあるので、これを利用してよいが、プログラムが簡単なのでプログラム例3ではこれを記号番地ADRから始まるサブブロックで計算させている。

なお、アセンブラプログラムにおける配列の取扱い、特にアレイサブテーブルの構成については上記マニュアルを参照されたい。

3. EXE\$の利用

前節の方法ではアセンブル時間に加えてFORT-RAN副プログラムのコンパイル時間が余計必要となる。そこで入出力ブロックを直接アセンブラプログラムの中で構成することを考えよう。

SDOS-43(II)FORTRANで書かれたソースプログラムをコンパイルすると、書式つき入出力文のREAD (u, f) k / WRITE (u, f) k はオブジェクトプログラムの中では次のように展開される。

```

CAL IO$R
DC   入出力の別----- { 入力 A 0100
                        { 出力 A 0101
DEF  入出力デバイスのアドレス
DEF  FORMAT文番号アドレス
    
```

```

LB   * + 3      { 0 単精度整数型
CAL LIST      { 1 単精度実数型
DC   データの型 { 2 倍精度実数型
DEF  データアドレス { 3 複素数型
                        { 4 論理型
                        { 5 倍精度整数型
    
```

```

CAL IO$E
    
```

3つのルーチンIO\$R, LI\$T, IO\$EはいずれもFORTRANオブジェクト実行時の補助サブブロックとしてのシステムモジュールEXE\$に属している。IO\$RとIO\$Eは1つのREAD文或いはWRITE文に対して1回だけコールされるが、LI\$Tは入出力並びの中の要素毎にコールされる。入力にカードリーダー、出力にラインプリンタを用いるとして、プログラム例4にこれらのルーチンの使い方を例示する。例ではIは単精度整数、Aは単精度実数、Dは倍精度実数、Cは複素数、Lは論理型、Jは倍精度整数を表わしている。

なお、FORMAT文を構成するストリングデータをDC命令で与えたのではコードが一致しないので、システムのIOCS引用ルーチンを用いてMOJI51およびMOJI61から始まる40語の中にそれぞれREAD文お

00001		SAP		
00002		SUB		MPRD
00003		COMMON		L(1),M(1),N(1)
00004	TOP	NO		
00005		CAL		INPUT
00006		DEF		AS
00007		DEF		BS
00008		DEF		CS
00009		L		L
00010		T		IEND
00011		L		CO
00012		T		I
00013	ILP	L		N
00014		T		JEND
00015		L		CO
00016		T		J
00017	JLP	FL		ZERO
00018		FT		W
00019		L		M
00020		T		KEND
00021		L		CO
00022		T		K
00023	KLP	TSJ		ADR
00024		DEF		L
00025		DEF		I
00026		DEF		K
00027		LB		INDX
00028		FL		B, A
00029		FT		AIK
00030		TSJ		ADR
00031		DEF		M
00032		DEF		K
00033		DEF		J
00034		LB		INDX
00035		FL		B, B
00036		FM		AIK
00037		FA		W
00038		FT		W
00039		MIS		K
00040		MDS		KEND
00041		J		KLP
00042		TSJ		ADR
00043		DEF		L
00044		DEF		I
00045		DEF		J
00046		FL		W
00047		LB		INDX
00048		FT		B, C
00049		MIS		J
00050		MDS		JEND
00051		J		JLP
00052		MIS		I
00053		MDS		IEND
00054		J		ILP
00055		CAL		OUTPUT
00056		DEF		AS
00057		DEF		BS
00058		DEF		CS
00059		J		I, TOP
00060	ADR	NO		
00061		LB		ADR
00062		L		J, 2
00063		M		J, 0
00064		XAR		
00065		A		J, 1
00066		M		C2
00067		XAR		
00068		T		
00069		J		B, 3
00070	C2	DC		2
00071	CO	DC		0
00072	ZERO	DC		0.0
00073	I	BRV		1.
00074	J	BRV		1
00075	K	BRV		1
00076	IEND	BRV		1
00077	JEND	BRV		1
00078	KEND	BRV		1
00079	INDX	BRV		1
00080	W	BRV		2
00081	AIK	BRV		2
00082	AS	DC		^1002
00083		DEF		A
00084		DC		18
00085		DC		3
00086		DC		3
00087	BS	DC		^1002
00088		DEF		B

行
列
番
号

```

00089      DC      18
00090      DC      3
00091      DC      3
00092      CS      DC      ^1002
00093      DEF     C
00094      DC      18
00095      DC      3
00096      DC      3
00097      A      BRV   18
00098      B      BRV   18
00099      C      BRV   18
00100      END
    
```

図3 プログラム例3

よびWRITE文に対するFORMAT文を読み込ませている。このルーチンのコーリングシーケンスの詳細については前記マニュアルを参照されたい。

FORTTRANプログラムの中では、READ/WRITE文に対するFORMATには必ず文番号をつけなければならないことになっているが、アセンブラプログラムの中では文番号そのものは必ずしもそれを要しないので、プログラム例では記号番地FMT51ならびにFMT61のところで文番号を実際に定義することなく単にNo operationとしている。

4. 付 言

IO\$,R, LI\$,T, IO\$,Eの3つのルーチンのコーリングシーケンスを導き出すのにSDOS-43(II)のモニタ、FORTTRANコンパイラおよびエグゼキュタを解読しなければならなかったが、IOCS引用ルーチンの使い方その他細々とした点に関する質問に対し、沖電気工業のソフトウェア事業部技術サポートセンタの幾人かの人達から懇切なる回答をお寄せいただいた。入出力に関する上記3つのルーチンがEXE\$のサブブロックだということも、ロードモジュールのダンプをとってみてはどうかという示唆によって知ることができた。記して謝意を表わしたい。

参 考 文 献

- 1) 森 亮一編 マイクロコンピュータ ハンドブック 朝倉 1979
- 2) 高等専門学校における情報処理教育の実態調査報告書 第2報, 高等専門学校情報処理教育研究協議会, 1977
- 3) OKITAC-4300 科学技術計算用ディスクオペレーティングシステム(II型)解説書

```

00001      SAP
00002      SUB      MAIN
00003      MAIN   NO
00004      TSJ     I, AF
00005      DC      ^0800
00006      DEF     MOJ151
00007      DC      80
00008      NO
00009      TSJ     I, AF
00010      DC      ^0800
00011      DEF     MOJ161
00012      DC      80
00013      NO
00014      CAL     IO$,R
00015      DC      ^0100
00016      DEF     DVC1
00017      DEF     FMT51
00018      LB     ++3
00019      CAL     LI$,T
00020      DC      0
00021      DEF     I
00022      LB     ++3
00023      CAL     LI$,T
00024      DC      1
00025      DEF     A
00026      LB     ++3
00027      CAL     LI$,T
00028      DC      2
00029      DEF     D
00030      LB     ++3
00031      CAL     LI$,T
00032      DC      3
00033      DEF     C
00034      LB     ++3
00035      CAL     LI$,T
00036      DC      4
00037      DEF     L
00038      LB     ++3
00039      CAL     LI$,T
00040      DC      5
00041      DEF     J
00042      CAL     IO$,E
00043      CAL     IO$,R
00044      DC      ^0100

00045      DEF     DVC2
00046      DEF     FMT61
00047      LB     ++3
00048      CAL     LI$,T
00049      DC      0
00050      DEF     I
00051      LB     ++3
00052      CAL     LI$,T
00053      DC      1
00054      DEF     A
00055      LB     ++3
00056      CAL     LI$,T
00057      DC      2
00058      DEF     D
00059      LB     ++3
00060      CAL     LI$,T
00061      DC      3
00062      DEF     C
00063      LB     ++3
00064      CAL     LI$,T
00065      DC      4
00066      DEF     L
00067      LB     ++3
00068      CAL     LI$,T
00069      DC      5
00070      DEF     J
00071      CAL     IO$,E
00072      DC      I, MAIN
00073      I      BRV   1
00074      J      BRV   2
00075      A      BRV   2
00076      D      BRV   3
00077      C      BRV   4
00078      L      BRV   1
00079      DVC1   DC     8
00080      DVC2   DC     1
00081      FMT51  NO
00082      MOJ151 BRV   40
00083      FMT61  NO
00084      MOJ161 BRV   40
00085      END
    
```

図4 プログラム例4