

プログラム開発環境の改善について

秦野 和郎・陳 的虎・竹松 英夫

Improvement of Programming Environment

Kazuo HATANO, Chen de HU and Hideo TAKEMATSU

Recently, we introduced a new personal computer PC-9801 VM2. This computer is controlled by the MS-DOS operating system. It is the first experience for us to work on the MS-DOS. To improve our programming environment, we must develop many utility programs. In this article, we introduce some useful utility programs coded by C programming language.

1. まえがき

パーソナルコンピュータが出現して既に相当な期間が経過している。この間のパーソナルコンピュータの発展は目覚ましくハードウェア、ソフトウェア共に初期の製品とは比較にならない位に高性能化してきている。

筆者らの研究室では従来16ビットパソコンMULTI16を主に使用してきたのであるが数年の頻繁な使用で老朽化し誤動作が目立つようになってきたので去年PC-9801VM2及びPC-98XAを購入した。これに伴って使用するOSをCP/M-86からMS-DOSに変更する事にした。

使用する計算機が変わるとそれまで使用してきたプログラムの多くが書き直しになるか或いは手直しを必要とする事になる。これは共同利用の大型計算機の場合と同じである。又、従来便利に利用してきた機能が新しい計算機には備わっていない事もある。このような事情から計算機の更新はなるべく避けたい。しかし技術革新の激しい今日、古くて遅い計算機をいつまでも使っているのは損である。そこで計算機を更新するのであるが、購入後しばらくの間は実用にならない。

共同利用の大型計算機の場合は更新する前にセンター職員なり然るべき委員会が導入前に更新による影響を少しでも少なくするように努力する。しかしパーソナルコンピュータの場合は不足する機能を自分で補なわなければならない。基本ソフトウェアであるOSやコンパイラを購入してもそれだけでは十分でない。

筆者らは数値計算のアルゴリズムの開発を専業としている。開発したアルゴリズムを検証するために専らFORTRANでプログラムを作っている。プログラム開発を効率的に行うためにはOSやコンパイラの他に種々

なユーティリティプログラムが必要である。筆者らは今回の機種変更に伴っていくつかのユーティリティプログラムを作製している。従来はそのようなユーティリティプログラムをFORTRANとアセンブラを使って書いていたのであるが、そのようにしては次の機種変更又はOSの変更時に又大巾な変更を必要とする。そこで今回は最近とみに評価の高まっている「C言語」で書く事にした。

本稿では作製中のいくつかのユーティリティプログラムについて述べる。本文中のプログラムはすべてMicroSoft社の「MS-C V. 4.0」の仕様に従って作製されている。

2. ソースファイルの番号つき印刷

使用するOSをCP/M-86からMS-DOSに変更して最初に困惑したのはMS-DOSではソースファイルを番号つきでディスプレイやプリンタに出力するコマンドが存在しない事であった。CP/M-86ではPIP(Peripheral Interchange Program)というかなり広い機能を有するコマンドがある。ファイル名が、MAIN, FORなるソースファイルを番号つきで表示するには

```
B>A: PIP CON: =MAIN, FOR[N]
```

とすればよい。又、プリンタに出力する際に60行毎にページ送りをし天地をあけるには

```
B>A: PIP PRN: =MAIN, FOR[NP]
```

とすればよい。

MS-DOSのもとでソースファイルを表示するには

```
B>COPY MAIN, FOR CON
```

とするのであるが番号を表示する機能はない。

そこでまず、図1に示すようなnuprinなるプログラムを作った。これは番号つきで複数個のソースファイル

```

Fri Jan 16 10:03:51 1987      NUPRIN_.BAT

1: TYPE NUPRIN_.BAT
2: SET PATH=C:\MSC\BIN
3: SET INCLUDE=C:\MSC\INCLUDE
4: SET LIB=C:\MSC\LIB
5: SET TMP=C:\
6: B:
7: CHDIR \NUPRIN
8: MSC B:NUPRIN,B:NUPRIN,CON;
9: LINK B:NUPRIN,B:NUPRIN;
10: B:NUPRIN B:\NUPRIN\NUPRIN.C

Fri Jan 16 10:03:59 1987      NUPRIN.C

1: /*****          **          *****/
2: /*   Print Files with Number  1- culumn only          */
3: /*   Usage  nuprin file1 file2 ...                    */
4: /*****          **          *****/
5: #include <stdio.h>
6: #include <time.h>
7:
8: main(argc, argv)
9: int argc;
10: char *argv[];
11: {   int i, j;
12:     long ltime;
13:     char buff[256] , bufft[26];
14:     FILE *fp;
15:
16:     for(j=1; j < argc; j++) {
17:         printf("%n");
18:         for(i = 1; i < 10; i++)   printf(" ");
19:         time(&ltime) ;
20:         sprintf(bufft, "%s" , ctime(&ltime)) ;
21:         bufft[24] = '\0' ; printf("%s",bufft);
22:         for(i = 1; i < 6 ; i++)   printf(" ");
23:         printf(" %s\n\n", argv[j]);
24:         if((fp = fopen(argv[j], "r")) == NULL) {
25:             printf("%7n Cannot Open File : %s\n\n"
26:                 ,argv[j]);
27:             continue;
28:         }
29:         i = 0 ;
30:         while((fgets(buff, 255, fp)) != NULL)
31:             printf( "%5d: %s", ++i, buff);
32:         fclose(fp);
33:     }
34: }

```

図 1

を表示するプログラムである。図 1 自体がこのプログラムで表示されている。nuprin_.bat 及び nuprin.c なる 2 つのソースファイルを表示するには

```
B>nuprin nuprin_.bat nuprin.c
```

と指示する。プリンタに印刷するには

```
B>nuprin nuprin_.bat nuprin.c>PRN
```

とする。

指定されたファイルの表示を開始する時点の日付と時刻及びファイル名を最初に表示する。nuprin.c の 19 行目

の time 及び 20 行目の ctime なる関数は MS-DOS から日付及び時刻を受け取り (time), 文字列の形に変換する (ctime) ものである。将来 OS を UNIX に変えるときこの部分のみは変更する必要があるかも知れない。

nuprin.c の 30 行目から 31 行目までがこのプログラムの主要部である。fgets なる関数で番号をつけ buff に一行分入力し printf なる関数で番号をつけて表示する。

最近の殆んどプリンターではファンフォールド紙の他にカットシート紙を使用する事ができる。ファンフォ

ールド紙に印刷する場合には左右に幾分か余白があるので紙を切り離れた後それを綴る事ができる。しかし、たとえば筆者らが使用している PC-PR101L ではカットシート紙に印刷するとき左端ぎりぎりの位置から印字するため綴じ代がない。このために COPY コマンドのような既製のコマンドでは不都合な事がある。nuprin では左端に綴じ代を作りたければ nuprin.c の 31 行目の %5d をたとえば %15d に変更すればよい。このような簡単なプログラムでもソースプログラムの形式で持っていると種々の状況に容易に対応する事ができる。

長いソースファイルを印刷し保管しておくには、たとえば 60 行毎に改頁し天地をあけると具合がよい。図 2 に示す paprin はそのような機能を持つプログラムである。後に示すプログラムはすべてこのプログラムで印刷している。

以上 2 つは CRT ディスプレー及び 80 桁のプリンタへの出力を想定したプログラムである。132 桁や 136 桁のプリンタに印刷するのにこれらのプログラムを使うと右半分が空白になってしまって紙が勿体ない。この場合には一枚に 2 列印刷するのが合理的である。プログラムのデバック中には一度になるべく多くの部分を見う事が好ましいのでこの面からも 2 列に印刷するのが好ましい。

図 3 は 136 桁のプリンタに複数個のソースファイルを印刷するプログラムである。このプログラムは 1 行が 60 桁を越える行を改行して出力するようにしているため幾分複雑である。プログラムの大雑把な流れは次のようになっている。doprin.c の 54 行目でまず、buffi に 1 行入力する。次に関数 copyli を呼び出して buffi の内容を buffp に移す。その際 buffi の内容が 60 字を越えているならそれに対応した措置をとる。buffp に 120 行分以上入ったら関数 oprint を呼び出してプリンタへ出力する。関数 oprint の主な操作部は 115 行目から 128 行目までである。この部分で buffp から buffo に移しながらプリンタへ出力する。

3. 文番号のつけ換え

殆んどのプログラム言語では、名札は文字列である。アセンブラですら名札は文字列である。従って名札に意味を持たせる事ができる。しかし FORTRAN のみは名札すなわち、文番号は 5 桁以内の自然数となっている。数字であるから意味を持たせる事はできない。文字列で与えることのできる名札はプログラムを見やすくするのに役立つ。名札は人がプログラムを見てわかりやすいようにきめればよいのである。

一方、FORTRAN の文番号は意味を持たせる事ができないのでプログラムをなるべく見やすくするためにど

のように文番号をきめるかについて種々の流儀がある。たとえば文献イ) では

- READ 文に対応する FORMAT 文には 500 から始まる 10 間隔の 3 桁の数
- WRITE 文に対応する FORMAT 文には 600 から始まる 10 間隔の 3 桁の数
- DO 文の文末文は 10 から始まる 10 隔の 2 桁の数
- GO TO 文などでその行に制御が移る文には 1 から始まる 1 間隔の 1 の位が 0 でない数 (通常 1 桁)

とつけている。

このような原則は 100 行程度迄のプログラムなら守る事ができるかも知れないが、一つのプログラム単位が 500 行とか 1000 行以上になると守る事が困難になる。又筆者らの経験では文番号の桁数は一定になっている方がプログラムをキーインするのに操作しやすい事がわかっている。そこで筆者らは文番号を 4 桁として第 2 欄～第 5 欄に書く事にしている。

文番号は定義される順 (第 2 ～第 5 欄に現われる順) に一定ずつ大きくなるようにするのがよい。でたらしめにつけると、たとえば、GO TO 1241 とあるとき行先はその文の上にあるのか下にあるのかわからない。従って文番号 1241 のついている行を始めから終りまでくまなくさがさなければならぬ。50 行程度の長さのプログラムならそれでもよいが長いプログラムになると大変である。そこで文番号をたとえば 1100 から始め 10 間隔で定義される順に大きくなるようにつける。

そのような方針でプログラムを作ってもデバッグの段階で、挿入、削除、置き換えをするためにプログラムが動くようになった時点では文番号の大小順が狂ってしまう。又最初 10 間隔でつけていた番号も 10 間隔ではなくなる。

プログラムが動くようになって答が出れば、それでそのプログラムの寿命は終りという場合はこれでよい。しかし多くの場合、一旦動くようになったプログラムは将来役に立つ可能性が大きい。又それを少し変更する事によって新しい機能のプログラムが出来る可能性もある。従って動くようになったプログラムは見易い形にして保存しておくべきである。そこで筆者らは文番号をつけ換えるプログラムを作って十数年来使っている。

図 4 に示すプログラム chgst は図 5 に示すようなプログラム (SLEQF2A1, OLD という名のファイル) を図 6 に示すようなプログラム (SLEQF2A1, FOR という名のファイルに入れる) に書き換える。図 5 と図 6 とでは文番号のみが異なる。図 5 のプログラムは文番号がでたらしめ

Fri Jan 16 10:05:10 1987 PAPERIN_.BAT

```

1: TYPE PAPERIN_.BAT
2: SET PATH=C:\MSC\BIN
3: SET INCLUDE=C:\MSC\INCLUDE
4: SET LIB=C:\MSC\LIB
5: SET TMP=C:\
6: B:
7: CHDIR \PAPERIN
8: MSC B:PAPERIN,B:PAPERIN,CON;
9: LINK B:PAPERIN,B:PAPERIN;
10: B:PAPERIN B:\PAPERIN\PAPERIN.C

```

Fri Jan 16 10:05:18 1987 PAPERIN.C

```

1: /*****          **          *****/
2: /*    Print Files with Number  1- column only    */
3: /*    Usage  paprin file1 file2 ...              */
4: /*****          **          *****/
5: #include <stdio.h>
6: #include <time.h>
7:
8: main(argc, argv)
9: int argc;
10: char *argv[];
11: {
12:     int i, il, j, line;
13:     long ltime ;
14:     char buff[256] , bufft[26];
15:     FILE *fp;
16:     line=0 ; printf("%014") ;
17:     for(j=1; j < argc; j++) {
18:         if( line <= 63 ) {
19:             printf("%n");
20:             for(il = 1; il < 10; il++) printf(" ");
21:             time(&ltime);
22:             sprintf(bufft, "%s" , ctime(&ltime));
23:             bufft[24] = '\0' ; printf("%s",bufft);
24:             for(il = 1; il < 6; il++) printf(" ");
25:             printf(" %s\n\n", argv[j]); line=line+3;
26:         }
27:         if((fp = fopen(argv[j], "r")) == NULL) {
28:             printf("%7s Cannot Open File : %s\n\n"
29:                 ,argv[j]); line=line+3;
30:             continue;
31:         }
32:         i = 0 ;
33:         while((fgets(buff, 255, fp)) != NULL) {
34:             if(line >= 63) {
35:                 printf("%014\n");
36:                 for(il=1 ; il<10 ; il++) printf(" ");
37:                 time(&ltime) ;
38:                 sprintf(bufft, "%s" , ctime(&ltime));
39:                 bufft[24]='\0' ; printf("%s", bufft);
40:                 for(il=1 ; il<6 ; il++) printf(" ");
41:                 printf(" %s\n\n", argv[j]); line=3;
42:             }
43:             printf( "%5d: %s", ++i, buff); line++ ;
44:         }
45:         fclose(fp);
46:     }
47: }

```

Fri Jan 16 10:06:29 1987 DOPRIN_.BAT

```

1: TYPE DOPRIN_.BAT
2: SET PATH=C:\MSC\BIN
3: SET INCLUDE=C:\MSC\INCLUDE
4: SET LIB=C:\MSC\LIB
5: SET TMP=C:\
6: B:
7: CHDIR %DOPRIN
8: MSC B:DOPRIN,B:DOPRIN,CON;
9: LINK B:DOPRIN,B:DOPRIN;
10: B:DOPRIN B:%DOPRIN%DOPRIN.C

```

Fri Jan 16 10:06:36 1987 DOPRIN.C

```

1: /*****          **          *****/
2: /*    Print Files with Number    2-  column print    */
3: /*    Usage    doprin file1 file2 ...    */
4: /*****          **          *****/
5: #include <stdio.h>
6: #include <time.h>
7:
8: char buffi[256], buffp[125][69], buffo[138] ;
9: char buffn[10] , fname1[20] , fname2[20] ;
10: int line, numb , init ;
11: FILE *fpt ;
12:
13: main(argc, argv)
14: int argc;
15: char *argv[];
16: {    int il, j , jl ;
17:     FILE *fp;
18:     line=0 ; init = 0 ;
19:     fpt = fopen ("PRN" , "w") ;
20:     for(j=1; j < argc; j++) {
21:         spacel() ;
22:         sprintf(fname2 , "%s\n" , argv[j]) ;
23:         for(il=0 ; il<40 ; il++) buffp[line][il]=' ' ;
24:         il = 39 ;
25:         while((buffp[line][il]=fname2[il-39]) != '\0')
26:             il++ ;
27:         line++ ;    spacel() ;
28:         for(il=0 , jl=0 ; fname2[il] != '\0' ; il++)
29:             if(fname2[il] != '\n')
30:                 fname2[jl++] = fname2[il] ;
31:             fname2[jl] = '\0' ;
32:         if(j == 1) {
33:             il = 0 ;
34:             while((fname1[il]=fname2[il]) != '\0')
35:                 il++ ;
36:         }
37:         if(line >= 120) {
38:             oprint() ; lclear() ;
39:         }
40:         if((fp = fopen(argv[j], "r")) == NULL) {
41:             spacel() ;
42:             sprintf(buffi," Cannot Open File : %s\n"
43:                 ,argv[j]);
44:             il = 0 ;
45:             while((buffp[line][il]=buffi[il]) != '\0')
46:                 il++ ;
47:             line++ ;    spacel() ;

```

Fri Jan 16 10:07:18 1987 DOPRIN.C

```

48:             if(line >= 120) {
49:                 oprint() ; lclear() ;
50:             }
51:             continue;
52:         }
53:         numb = 0 ;
54:         while((fgets(buffi, 255, fp)) != NULL) {
55:             numb++ ; copyli() ;
56:             if(line >= 120) {
57:                 oprint() ; lclear() ;
58:             }
59:         }
60:         fclose(fp);
61:     }
62:     if(line > 0) oprint() ;
63:     fprintf(fpt , "%n%n");
64:     fclose (fpt) ;
65: }
66:
67: spacel()
68: {   buffp[line][0] = '\n' ; buffp[line][1] = '\0' ;
69:     line++ ;
70: }
71:
72: copyli()
73: {   int j1 , j2 , k1 , c ;
74:     sprintf(buffn , "%5d: " , numb) ;
75:     for(j2=0 ; j2<7 ; j2++) buffp[line][j2]=buffn[j2] ;
76:     j1 = 0 ; k1 = 0 ;
77:     while((buffp[line][j1-k1+7] = buffi[j1])!= '\0') {
78:         j1++ ; c=buffi[j1] ;
79:         if(j1%60 == 0 && c != '\n' && c != '\0') {
80:             buffp[line][67]='\n' ;
81:             buffp[line][68]='\0' ;
82:             line++ ; k1+=60 ;
83:             for(j2=0 ; j2<7 ; j2++)
84:                 buffp[line][j2]=' ' ;
85:         }
86:     }
87:     line++ ;
88: }
89:
90: oprint()
91: {   int i1 , i2 , i9 , j1 ;
92:     long ltime ;
93:     char bufft[26] ;
94:     if(init == 0)
95:         fprintf(fpt , "%n") ;
96:     else
97:         fprintf(fpt , "%014%n") ;
98:     init++ ;
99:     for(j1=1 ; j1<10 ; j1++) fprintf(fpt , " ") ;
100:    time(&ltime);
101:    sprintf(bufft , "%s" , ctime(&ltime));
102:    bufft[24] = '\0' ; fprintf(fpt , "%s" , bufft);
103:    for(j1=1 ; j1<6 ; j1++) fprintf(fpt , " ") ;
104:    fprintf(fpt , "%s" , fname1) ;
105:    if(line > 60) {
106:        for(j1=0 ; j1<31-strlen(fname1) ; j1++)
107:            fprintf(fpt , " ") ;

```

```

Fri Jan 16 10:08:06 1987      DOPRIN.C

108:         for(j1=1 ; j1<10 ; j1++) fprintf(fpt , " ");
109:         fprintf(fpt , "%s" , bufft);
110:         for(j1=1 ; j1<6 ; j1++) fprintf(fpt , " ");
111:         fprintf(fpt , "%s¥n¥n" , fname2) ;
112:     }
113:     else
114:         fprintf(fpt , "¥n¥n") ;
115:     for(il=0 ; il<60 ; il++) {
116:         if(il >= line) return ;
117:         i2 = il + 60 ; i9 = 0 ;
118:         while((buffo[i9]=buffp[i1][i9]) != '¥0')
119:             i9++ ;
120:         if(i2 < line) {
121:             for(j1=i9-1 ; j1<69 ; j1++)
122:                 buffo[j1] = ' ' ;
123:             i9 = 0 ;
124:             while((buffo[i9+69]=buffp[i2][i9]) != '¥0')
125:                 i9++ ;
126:         }
127:         fprintf(fpt , "%s" , buffo) ;
128:     }
129:     il = 0 ;
130:     while((fname1[i1]=fname2[i1]) != '¥0')
131:         i1++ ;
132: }
133:
134: lclear()
135: {   int klin , il , j1 ;
136:     if(line > 120) {
137:         klin = line-120 ;
138:         for(il=0 ; il<klin ; il++) {
139:             while((buffp[i1][j1]=buffp[i1+120][j1])
140:                 != '¥0') j1++ ;
141:         }
142:     }
143:     line -= 120 ;
144: }

```

図 3-3

であるから見にくい。GO TO 8213 という文を見たとき次はどこへ制御が移るかを知るのに苦労する。この程度の長さならまだよいか500行程度以上のプログラムになると困ってしまう。図6では文番号が規則的に並んでいるので、たとえばGO TO 1170 という文を見たとき次に制御がどこへ行くかを簡単に見出す事ができる。

図4に示すプログラムでは文番号をつけ換えるべきプログラムが入っているファイルのファイル名をchgst.cの25行目で読み取り、文番号をつけ換えたプログラムの入るファイルのファイル名を27行目で読む。次に29行目で新たにつけるべき最初の文番号を読み(4桁)31行目でいくつの間隔で文番号をつけるかを与える(2桁)。41行目から68行目までは文番号を抽出しそれらを文番号表noldに登録する。69行目から77行目までで新しい文番号表nnewを作る。79行目から131行目までで文番号を更新

して新しいファイルを作る。

FORTRAN が新しい規格, FORTRAN77 になり, ブロック IF 文を使えるようになって必要な文番号の個数は激減した。又, 紙カードの使用からファイルの使用に変わって FORTRAN のプログラムでも「字下げ」を普通に行うようになった。そのような事情から chgst なるプログラムの必要性もかなり薄らいで来ている。それでも FORTRAN が使われる限りこのプログラムは有用である。

4. プロットファイルの出力

かつて計算機はカードリーダーからプログラムやデータを読み込んで計算しラインプリンタへ結果を出力するものであった。しかし XYプロッターやグラフィックデ

Fri Jan 16 10:09:05 1987 CHGST_.BAT

```

1: SET PATH=C:\MSC\BIN
2: SET INCLUDE=C:\MSC\INCLUDE
3: SET LIB=C:\MSC\LIB
4: SET TMP=C:\
5: B:
6: CHDIR \CHGST
7: MSC B:CHGST,B:CHGST,CON;
8: LINK B:CHGST,B:CHGST /STACK:20000;
9: B:CHGST

```

Fri Jan 16 10:09:14 1987 CHGST.C

```

1: /*****          **          *****/
2: /*      Change the Statement Numbers (FORTRAN)      */
3: /*      Usage          chgst [-p]                    */
4: /*****          **          *****/
5: #include <stdio.h>
6: #include <ctype.h>
7:
8: main(argc, argv)
9: int  argc ;
10: char *argv[] ;
11: { int nold[2000] , nnew[2000] , itabp[2][100];
12:   int inst , intv , ipn1 , ipn0 , kosp , kosl , ipn2;
13:   int i1 , i2 , numb , kpl , ic , icl , irl , exit1 ;
14:   int kprin , line , kchg ;
15:   char file1[64] , file2[64] , buff[82] , jcn[5] ;
16:   char *s1 ;
17:   FILE *fp1 , *fp2 ;
18:   kprin = 0 ;
19:   if(--argc > 0 && (*++argv)[0] == '-') {
20:     s1 = argv[0]+1 ;
21:     if( *s1 == 'p' || *s1 == 'P' ) kprin = 1 ;
22:   }
23:   for( ; ; ) {
24:     printf("          Source File=");
25:     scanf("%s",file1);
26:     printf("          Destination File=");
27:     scanf("%s",file2);
28:     printf("Initial Statement Number(4d)=");
29:     scanf("%4d", &inst);
30:     printf("          Interval(2d)= ");
31:     scanf("%2d", &intv);
32:     if((fp1 = fopen(file1, "r")) == NULL) {
33:       printf("Missing Source File %s\n");
34:       continue ;
35:     }
36:     if((fp2 = fopen(file2, "w")) == NULL) {
37:       printf("Cannot create Destination File%s\n");
38:       break ;
39:     }
40:     /* --- pick up the statement numbers -----*/
41:     ipn1 = 0 ; kosp = 0 ;
42:     ipn0 = ipn1 ; kosl = 0 ;
43:     while((fgets(buff, 81, fp1)) != NULL) {
44:       if(buff[0] == 'C' ||
45:          (buff[5] != ' ' && buff[5] != '0'))
46:         continue ;
47:       exit1 = 0 ;
48:       for(i1=0 ; i1<=3 ; i1++) {

```


Fri Jan 16 10:09:59 1987 CHGST.C

```

49:             if( isdigit(buff[il+1]) )
50:                 jcn[il] = buff[il+1] ;
51:             else if(buff[6] == 'E' &&
52:                 buff[7] == 'N' &&
53:                 buff[8] == 'D' &&
54:                 buff[9] == '¥n' ) {
55:                 itabp[0][kosp] = ipn0 ;
56:                 itabp[1][kosp] = kosl ;
57:                 kosp++ ; ipn0 = ipn1 ; kosl = 0 ;
58:                 exit1 = 1 ; break ;
59:             }
60:             else {
61:                 exit1 = 1 ; break ;
62:             }
63:         }
64:         if( exit1 ) continue ;
65:         jcn[il]='¥0' ;
66:         sscanf(jcn , "%4d" , &numb);
67:         nold[ipn1]=numb ; ipn1++ ; kosl++ ;
68:     }
69:     for(kp1=0 ; kp1<=kosp-1 ; kp1++) {
70:         ipn0 = itabp[0][kp1]; kosl = itabp[1][kp1];
71:         if( kosl != 0 ) {
72:             for ( il=0 ; il<=kosl-1 ; il++) {
73:                 i2 = ipn0 + il ;
74:                 nnew[i2]=inst + intv * il ;
75:             }
76:         }
77:     }
78: /* --- change the statement numbers -----*/
79:     kosp = 0 ; rewind( fp1 ) ; line = 0 ;
80:     ipn0 = itabp[0][kosp] ; kosl = itabp[1][kosp] ;
81:     ipn2 = ipn0 + kosl ;
82:     while((fgets(buff, 81, fp1)) != NULL) {
83:         line++ ; kchg = 0 ;
84:         if(buff[0] == 'C') {
85:             fputs(buff, fp2) ; continue ;
86:         }
87:         if(buff[6] == 'E' && buff[7] == 'N' &&
88:             buff[8] == 'D' && buff[9] == '¥n' ) {
89:             fputs(buff, fp2) ; kosp++ ;
90:             ipn0 = itabp[0][kosp] ;
91:             kosl = itabp[1][kosp] ;
92:             ipn2 = ipn0 + kosl ; continue ;
93:         }
94:         if( kosl == 0 ) {
95:             fputs(buff, fp2) ; continue ;
96:         }
97:         ic = 0 ; ir1 = strlen(buff) ;
98:         while(ic+4 <= ir1) {
99:             ic++ ; exit1 = 0 ;
100:            for (il = 0; il <= 3; il++) {
101:                icl = ic + il ;
102:                if( isdigit(buff[icl]) )
103:                    jcn[il] = buff[icl] ;
104:                else {
105:                    exit1 = 1 ; break ;
106:                }
107:            }
108:            if( exit1 ) continue ;

```

Fri Jan 16 10:10:56 1987 CHGST.C

```

109:             jcn[i1] = '¥0' ;
110:             if(ic != 1)
111:                 if( isdigit(buff[ic-1]) ||
112:                    isdigit(buff[ic+4]) )
113:                     continue ;
114:             sscanf(jcn , "%4d" , &numb) ;
115:             for (i1 = ipn0 ; i1<=ipn2-1 ; i1++) {
116:                 ipn1 = i1 ;
117:                 if(numb == nold[i1]) {
118:                     numb = nnew[i1] ;
119:                     sprintf(jcn , "%4d" , numb) ;
120:                     for (i2=0 ; i2<=3 ; i2++) {
121:                         ic1 = ic + i2 ;
122:                         buff[ic1] = jcn[i2] ;
123:                     }
124:                     ic += 3 ; kchg = 1 ; break ;
125:                 }
126:             }
127:         }
128:         fputs(buff , fp2) ;
129:         if( kchg && kprin )
130:             printf("%5d: %s" , line , buff) ;
131:     }
132:     fclose( fp1 ) ; fclose( fp2 ) ;         ipn1++ ;
133:     printf(" Number of -END- statement=%4d¥n"
134:           , kosp) ;
135:     printf(" Number of statement number=%4d¥n"
136:           , ipn1) ;
137:     printf("¥n¥n") ;
138: }
139: }

```

図4-3

ィスプレーの低廉化に伴って計算結果をグラフ等、図形の形で出力する事が普及してきた。パソコンは出現当初から図形出力機能を持つのが当然と考えられていたようである。但しそれはBASICという個々のパソコン毎の閉じた処理系の上で利用しうるに過ぎなかった。CP/MやMS-DOSのような開いた処理系、すなわち利用者の意思でソフトウェアを追加してゆけるような環境では図形出力のような機能は提供されていなかった。必要ならば利用者自身で構成せねばならなかったのである。

しかし最近その間の事情が少しずつ変わってきている。

現在発売されているFORTRAN処理系にはGraphic Libraryを含むか或いは別売りで入手できるものがある。NEC製のPC-FORTRANにはCRTディスプレイに面像を出力するライブラリが含まれている。ライフボード社が発売しているPro FORTRAN77には別売りのライブラリとしてGKS仕様のGraphic Library、カルコン仕様のプロッタールーチンがある。従って図形出力、図形処理をFORTRANで行う事が以前に較べると非常に楽になった筈である。

しかしPC-FORTRANはそれ自体使いにくい処理系であるし、Pro FORTRAN77用のプロッタールーチンも使い易いライブラリではないように思われる。

筆者らは使い易いという事でMicro-Soft社のMS-FORTRANを使っている。MULTI16ではVersion 3.13を使っていたがPC-9801に変わってからはVersion 3.31を使っている。パソコンを変えた時点でOSがCP/M-86からMS-DOSに変わったがどちらもMS-FORTRANを使っているのでMULTI16で作ったFORTRANのプログラムは全く無修正でPC-9801で動く。MULTI16で作ったFORTRANのプログラムが豊富にあるので今更別のFORTRAN処理系に切り換える訳にゆかない。

MS-FORTRANには図形出力や図形処理に関するライブラリは何もない。そこでとりあえず最低限必要な図形出力の機能としてカルコン仕様のプロッタールーチンによるプロッターへの出力とCRTディスプレイへの出力が要求された。

筆者らは先にFM-7用のプロッタールーチンを作製

Fri Jan 16 10:00:55 1987 SLEQF2A1.OLD

```

1: C*
2: C***** SLEQF2 ** SLEQF2A1 *****
3: C*
4: C*          SOLUTION OF SIMULTANUOUS LINEAR EQUATION
5: C*          GAUSSIAN ELIMINATION (WITH PARTIAL PIVOTTING)
6: C*
7: C*****          **          *****
8:          SUBROUTINE SLEQF2(A,B,N,ND)
9:          IMPLICIT INTEGER*4 (I-N)
10:         IMPLICIT REAL *4 (A-H,O-Z)
11:         DIMENSION A(ND,20),B(20)
12:         IF(N.GT.1) GO TO 1207
13:         B(1)=B(1)/A(1,1)
14:         RETURN
15: 1207 DO 1194 IP=1,N
16:         AMAX=0.0EQ
17:         DO 3124 IR=IP,N
18:         IF(ABS(A(IR,IP)).LT.AMAX) GO TO 3422
19:         AMAX=ABS(A(IR,IP))
20:         IR1=IR
21: 3422     CONTINUE
22: 3124     CONTINUE
23:         IF(IP.EQ.IR1) GO TO 2421
24:         DO 1153 IC=IP,N
25:         AMAX=A(IP,IC)
26:         A(IP,IC)=A(IR1,IC)
27:         A(IR1,IC)=AMAX
28: 1153     CONTINUE
29:         AMAX=B(IP)
30:         B(IP)=B(IR1)
31:         B(IR1)=AMAX
32: 2421     IF(IP.EQ.N) GO TO 8213
33:         IP1=IP+1
34:         DO 7269 IR=IP1,N
35:         DEL=A(IR,IP)/A(IP,IP)
36:         DO 6532 IC=IP,N
37:         A(IR,IC)=A(IR,IC)-A(IP,IC)*DEL
38: 6532     CONTINUE
39:         B(IR)=B(IR)-B(IP)*DEL
40: 7269     CONTINUE
41: 8213     CONTINUE
42: 1194     CONTINUE
43:         NP1=N+1
44:         DO 6916 IPD=1,N
45:         IP=NP1-IPD
46:         SUM=B(IP)
47:         IF(IP.EQ.N) GO TO 3421
48:         IP1=IP+1
49:         DO 6123 IC=IP1,N
50:         SUM=SUM-A(IP,IC)*B(IC)
51: 6123     CONTINUE
52: 3421     B(IP)=SUM/A(IP,IP)
53: 6916     CONTINUE
54:         RETURN
55:         END

```

Fri Jan 16 10:01:53 1987

SLEQF2A1.FOR

```

1: C*
2: C***** SLEQF2 ** SLEQF2A1 *****
3: C*
4: C*          SOLUTION OF SIMULTANUOUS LINEAR EQUATION          *
5: C*          GAUSSIAN ELIMINATION (WITH PARTIAL PIVOTTING)    *
6: C*
7: C*****          **          *****
8:          SUBROUTINE SLEQF2(A,B,N,ND)
9:          IMPLICIT INTEGER*4 (I-N)
10:         IMPLICIT REAL    *4 (A-H,O-Z)
11:         DIMENSION A(ND,20),B(20)
12:         IF(N.GT.1) GO TO 1100
13:         B(1)=B(1)/A(1,1)
14:         RETURN
15: 1100 DO 1180 IP=1,N
16:         AMAX=0.0E0
17:         DO 1120 IR=IP,N
18:             IF(ABS(A(IR,IP)).LT.AMAX) GO TO 1110
19:             AMAX=ABS(A(IR,IP))
20:             IR1=IR
21: 1110     CONTINUE
22: 1120     CONTINUE
23:         IF(IP.EQ.IR1) GO TO 1140
24:         DO 1130 IC=IP,N
25:             AMAX=A(IP,IC)
26:             A(IP,IC)=A(IR1,IC)
27:             A(IR1,IC)=AMAX
28: 1130     CONTINUE
29:             AMAX=B(IP)
30:             B(IP)=B(IR1)
31:             B(IR1)=AMAX
32: 1140     IF(IP.EQ.N) GO TO 1170
33:             IP1=IP+1
34:             DO 1160 IR=IP1,N
35:                 DEL=A(IR,IP)/A(IP,IP)
36:                 DO 1150 IC=IP,N
37:                     A(IR,IC)=A(IR,IC)-A(IP,IC)*DEL
38: 1150             CONTINUE
39:                 B(IR)=B(IR)-B(IP)*DEL
40: 1160             CONTINUE
41: 1170             CONTINUE
42: 1180             CONTINUE
43:             NP1=N+1
44:             DO 1210 IPD=1,N
45:                 IP=NP1-IPD
46:                 SUM=B(IP)
47:                 IF(IP.EQ.N) GO TO 1200
48:                 IP1=IP+1
49:                 DO 1190 IC=IP1,N
50:                     SUM=SUM-A(IP,IC)*B(IC)
51: 1190             CONTINUE
52: 1200             B(IP)=SUM/A(IP,IP)
53: 1210             CONTINUE
54:             RETURN
55:             END

```

Fri Jan 16 10:12:00 1987 FIPL16_.BAT

```

1: TYPE FIPL16_.BAT
2: SET PATH=C:\MSC\BIN
3: SET INCLUDE=C:\MSC\INCLUDE
4: SET LIB=C:\MSC\LIB
5: SET TMP=C:\
6: B:
7: CHDIR \FIPL16
8: MSC B:FIPL16,B:FIPL16,CON;
9: LINK B:FIPL16,B:FIPL16;
10: B:FIPL16

```

Fri Jan 16 10:12:10 1987 FIPL16.C

```

1: /*****          **          *****/
2: /* Plotting of Plotter File */
3: /* Usage fipl16 */
4: /*****          **          *****/
5: #include <stdio.h>
6:
7: main()
8: { int ixl , iyl , height , speed , spol ;
9:   float xorg, yorg, factor ;
10:   char file[64] , buff[81] , kcom;
11:   FILE *fp , *fpp ;
12:
13:   for( ; ; ) {
14:     printf("\n Source File =");
15:     scanf("%s",file);
16:     if((fp = fopen(file, "r")) == NULL) {
17:       printf("\n Missing Source File %s\n",
18:             file);
19:       continue;
20:     }
21:     fpp = fopen ("PRN" , "w") ;
22:     printf(" Origin x,y =");
23:     scanf("%f %f",&xorg,&yorg);
24:     printf(" Factor ="); scanf("%f",&factor);
25:     printf(" Speed (0-10)="); scanf("%d",&speed );
26:     xorg = xorg * 100.0 ; yorg = yorg * 100.0 ;
27:     if( speed < 0 ) speed = -speed ;
28:     if(speed != 0) speed = (speed-1) % 10 + 1 ;
29:     fprintf(fpp, " H\n");
30:     fprintf(fpp, " T%d\n", speed);
31:     for( ; ; ) {
32:       if((fgets(buff, 82, fp)) == NULL) {
33:         fflush( fpp ); break;
34:       }
35:       kcom = buff[0];
36:       switch (kcom) {
37:         case 'M':
38:           sscanf(buff,"M%4d,%4d", &ixl,&iyl);
39:           ixl = xorg + ixl * factor ;
40:           iyl = yorg + iyl * factor ;
41:           fprintf(fpp," M%4d,%4d\n",ixl,iyl);
42:           break ;
43:         case 'D':
44:           sscanf(buff,"D%4d,%4d", &ixl,&iyl);
45:           ixl = xorg + ixl * factor ;
46:           iyl = yorg + iyl * factor ;
47:           fprintf(fpp," D%4d,%4d\n",ixl,iyl);

```

```

Fri Jan 16 10:12:54 1987      FIPL16.C

48:             break ;
49:             case 'S':
50:                 sscanf(buff, "S%2d", &height);
51:                 height = height * factor ;
52:                 fprintf(fpp, " S%2d¥n", height);
53:                 break ;
54:             case 'T':
55:                 sscanf(buff, "T%2d", &spol);
56:                 if(speed == 0)
57:                     fprintf(fpp, " T%2d¥n", spol);
58:                 break ;
59:             default :
60:                 fprintf(fpp, " %s", buff);
61:                 break ;
62:         }
63:     }
64:     fclose(fp);
65:     fclose(fpp);
66: }
67: }

```

図7-2

```

Fri Jan 16 10:13:38 1987      GRPL16_.BAT

1: TYPE GRPL16_.BAT
2: SET PATH=C:¥MSC¥BIN
3: SET INCLUDE=C:¥MSC¥INCLUDE
4: SET LIB=C:¥MSC¥LIB
5: SET TMP=C:¥
6: B:
7: CHDIR ¥GRPL16
8: MSC B:GRPL16,B:GRPL16,CON;
9: LINK B:GRPL16,B:GRPL16;
10: B:GRPL16

```

```

Fri Jan 16 10:13:49 1987      GRPL16.C

1: /*****          **          *****/
2: /*  Display  of  Plotter  File          */
3: /*  Usage    grpl16                    */
4: /*****          **          *****/
5: #include <stdio.h>
6:
7: main()
8: {   int   ix1 , iy1 , ix2 , iy2 , kpen
9:     ,fx1 , fyl , fx2 , fy2      ;
10:   float conx1, cony1, factor ;
11:   char  file[64] , buff[81] , kcom;
12:   FILE *fp;
13:
14:   for( ; ; ) {
15:       printf("¥n Source File =");
16:       scanf("%s",file);
17:       if((fp = fopen(file, "r")) == NULL) {
18:           printf("¥7¥n Missing Source File  %s¥n¥n"
19:               ,file);
20:           continue;
21:       }

```

図8-1

```

22:         printf(" Factor      =");
23:         scanf("%f", &factor);
24:         conx1 = (639.0/3600.0) * factor;
25:         cony1 = (399.0/2700.0) * factor;  kpen = 7 ;
26:         printf("¥033*¥033=  ");
27:         printf("¥033¥007init¥n");
28:         printf("¥033¥007screen 3,0,0,1¥n");
29:         printf("¥033¥007roll 399,0,1¥n");
30:         printf("¥033¥007roll 1,0,1¥n");
31:         for( ; ; ) {
32:             if((fgets(buff, 82, fp)) == NULL) {
33:                 printf("¥7¥7¥7¥7¥7");
34:                 scanf("%s", buff);          break;
35:             }
36:             kcom = buff[0];
37:             switch (kcom) {
38:                 case 'M':
39:                     sscanf(buff, "M%4d,%4d",&ix1,&iy1);
40:                     break ;
41:                 case 'D':
42:                     sscanf(buff, "D%4d,%4d",&ix2,&iy2);
43:                     fx1 =          conx1 * ix1 ;
44:                     fy1 = 399.0 - cony1 * iy1 ;
45:                     fx2 =          conx1 * ix2 ;
46:                     fy2 = 399.0 - cony1 * iy2 ;
47:                     printf("¥033¥007line ");

                Fri Jan 16 10:14:33 1987          GRPL16.C

48:                     printf("%d,%d,%d,%d,%d,%d¥n"
49:                             ,fx1,fy1,fx2,fy2,kpen,0);
50:                     ix1 = ix2 ;    iy1 = iy2 ;
51:                     break ;
52:                 case 'J':
53:                     sscanf(buff , "J%1d", &kpen);
54:                     kpen = (8 - kpen % 8) % 8 ;
55:                     break ;
56:                 default :
57:                     break ;
58:             }
59:         }
60:         fclose(fp);
61:     }
62: }

```

図8-2

し、その概要を報告した。この時には富士通製の8ビットパソコン、FM-7とグラフィック社のマイプロットジュニアを有効に使うためにカルコン仕様のプロッタールーチンをFORTRAN-80で作った。その後、同じ考えでMULTI16とマイプロットIIのためにカルコン仕様のプロッタールーチンをMS-FORTRANで作ってこれまで使用してきた。これは次のような3つのプログラムから成っている。

1) PLOT, SYMBOL, NUMBERなどのルーチンが呼ばれるとそれに応じたマイプロットII用のコマンドを、PLOTSで指定されたファイルに出力するサブルーチン群。MS-FORTRANで書かれている。

2) マイプロットII用コマンドの出力されたファイルすなわちプロットファイルからコマンドを読み出してプロッターへ出力するプログラム。プロッター上でのどの位置から書き始めるかを指定できる。又、尺度を指定し縮小して書いたり拡大して書いたりする事ができる。MS-FORTRANとマクロアセンブラで書かれている。

3) プロットファイルからコマンドを呼び出してCRTディスプレイ上に図形を出力するプログラム。尺度を指定する事ができる。MS-FORTRANとマクロアセンブラを使って書かれている。

今回PC-9801に変更するに当たって2)のプログラムと

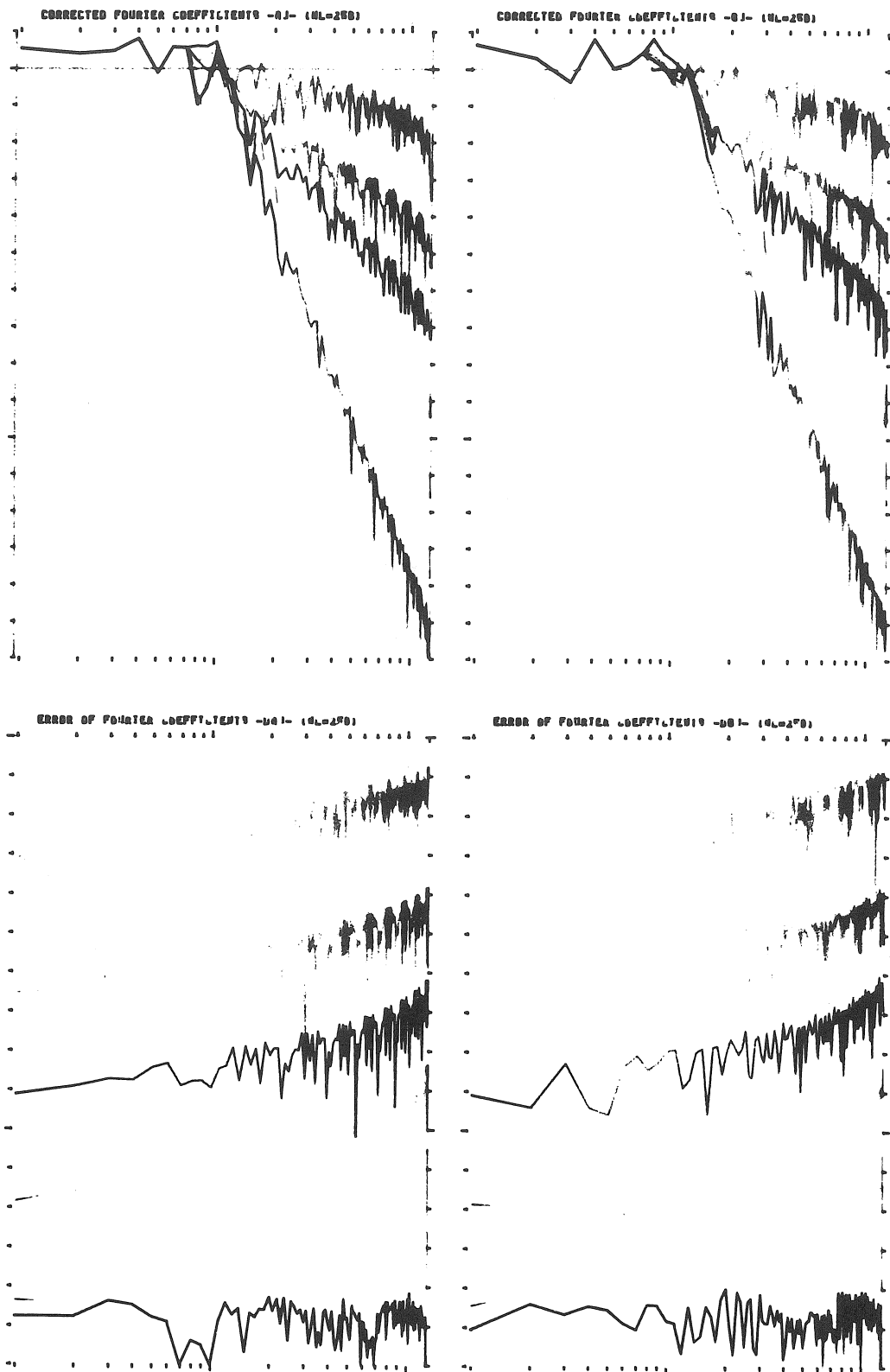


图 9

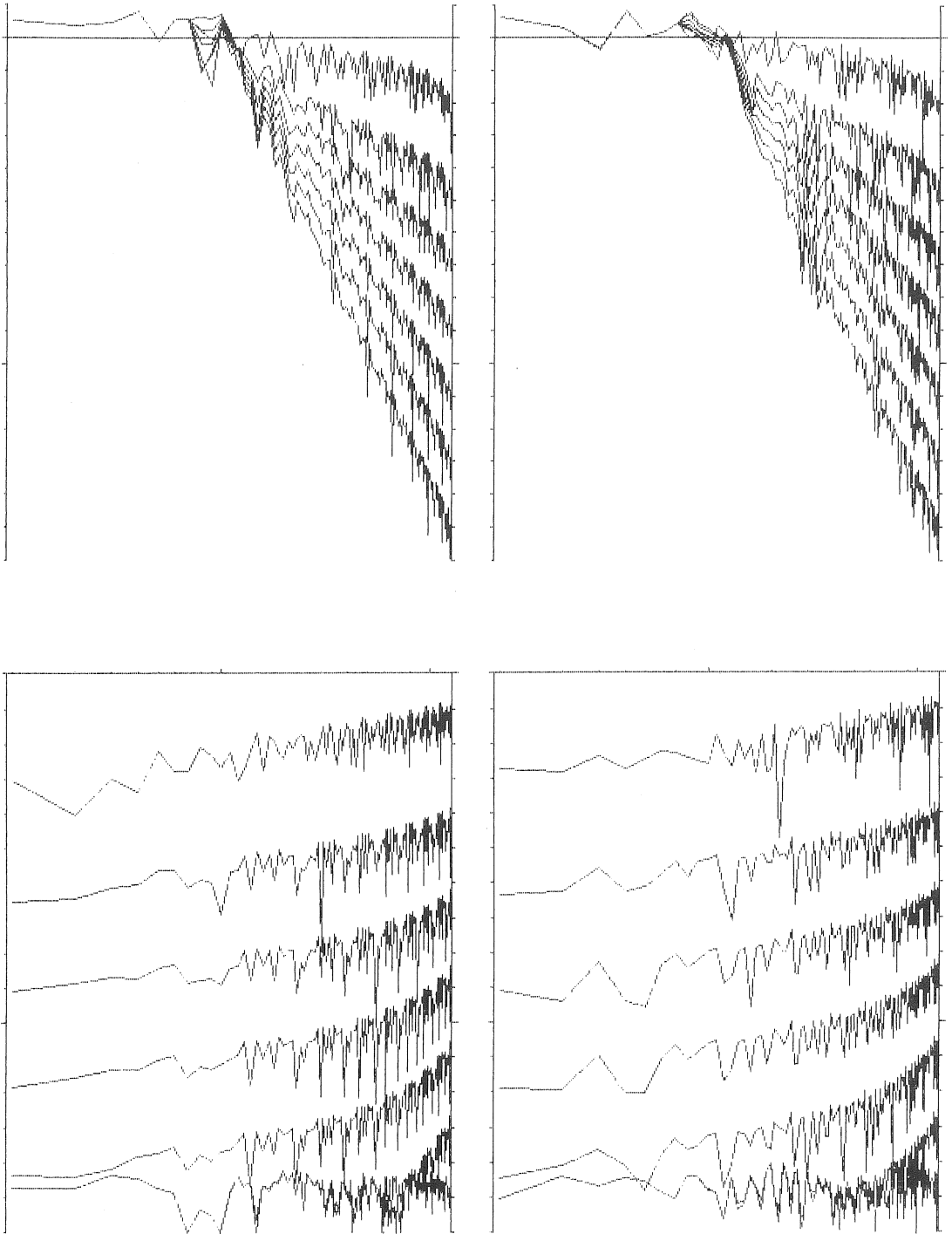


図10

3) のプログラムを「言語C」で書いた。1) のプログラムはMS-FORTRANのみで書かれているので無修正で動いた。従って今回は手を加える必要はなかった。

2) のプログラムを図7に示す。使用しているマイプロットIIはセントロニクスインターフェース付きであるからPRNへ出力すればよい。プロットのファイルからコマンドを読み出してプロッタへ出力するのであるが、位置、尺度に関連するコマンドは引数の大きさを計算して計算された値を出力する(Mコマンド、Dコマンド)。文字の高さを指定するコマンド(Sコマンド)も高さを変えて出力する。

3) のプログラムすなわちプロットファイルをCRTディスプレイへ出力するプログラムを図8に示す。このプログラムではカノーブス社のEGR98を使用している。grpl16を走らせるためにはあらかじめEGR98を実行しておかなければならない。

2) のプログラムすなわちfipl16で書いた例を図9に、3) のプログラムすなわちgrpl16でCRTディスプレイ上に書いて、コピー機能によりプリンターへコピーした例を図10に示す。

5. むすび

パソコンは優れたプログラム開発環境を提供してくれる。標準的な入出力装置についてはOSにはほぼ十分な機

能が含まれている。従ってその範囲内でも有用であるが、本稿で述べたような機能を追加する事で更に使い易くなる。ソースファイルの印刷のような一見つまらないように思われる事でも本稿で紹介したようなユーティリティプログラムを作ると非常に便利である。

応用によっては十分な機能が備わっていない事はパソコンの短所であるが、逆に利用者の負担と努力によって、よりよい環境を作り出してゆける事はパソコンの長所であると思う。今後種々な周辺装置が開発され我々利用者に提供されると思われる。そのような新しい周辺装置を有効に生かすには利用者側の努力が不可欠である。

本稿ではいくつかのユーティリティプログラムを紹介したが筆者らもこれだけでは十分な環境とっていない。今後もこれらを改良し又、種々なプログラムを開発してゆく予定である。

参考文献

- 1) 森口繁一：JIS FORTRAN 入門 [上], [下] 第3版, 東京大学出版会, 東京, 1984.
- 2) 秦野和郎・葉建勲・竹松英夫：FM-7用プロッターチェーンの作製, 愛知工業大学研究報告No.19 (1984) pp.51-59

(受理 昭和62年1月25日)