



INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE



Introduction to FLEX

H4.SMR. 403/9

FIFTH COLLEGE ON MICROPROCESSORS: TECHNOLOGY AND APPLICATIONS
IN PHYSICS

2- 27 October 1989

September 22, 1989

Introduction to FLEX

C. VERKERK
CERN, DD Division, Geneva, Switzerland

C. Verkerk
CERN
Geneva

FLEX

These notes are intended for internal distribution only.

ROSY Commands

- **Inspect/Modify Memory:** *M-COMPARE, M-DUMP, M-FILL, M-LIST, M-LOAD, M-MODIFY, M-VERIFY, COPY, SEARCH*
- **Inspect/Modify Registers:** *REG-DISPLAY, REG-MODIFY*
- **Debugging:** *BR-CLEAR, BR-SET, DECODE, SYMBOLS, TRACE*
- **Execution/I-O:** *GO, RESTART, MON-EXEC, SLAVE, TMODE, FLEX, OS-9*
- **Miscellaneous:** *HELP, DATA-CONVERT, SET-NUMBER-BASE*

Monitor Calls

- User calls to routines of the **monitor**. Also used by the monitor itself.
 - Character/Text Input/Output.
 - Number Input/Output.
 - Setting of vectors, I/O parameters
 - Error messages.
 - Task activation/de-activation, clock.

ROSY vs. FLEX

- ROSY monitor does **not** have:
 - File System,
 - Editor,
 - Assembler,
 - High-level Language Processors.
- These services are provided by **FLEX**.
FLEX makes use of ROSY routines:
 - Character Input/Output from/to terminal,
 - Disk driver routines,
 - Task activation for printer spooling.

How to get into FLEX

- How do we get into FLEX?
 - Switch on (or reset)
 - Wait for: **ROSY>**
 - Then type: **FLEX**, (or **F**)
 - System asks for date
 - Type in **MM DD YY** (e.g. 10 04 89)
 - After a few seconds appears:
+ + +

ROSY >f ← START UP OF FLEX

6809 FLEX V3.01

DATE (MM,DD,YY) ? 09 29 86

ASIAN REGIONAL COLLEGE ON MICROPROCESSORS

HEFEI - CHINA

22 SEPTEMBER - 17 OCTOBER 1986

+++ date ←

SEPTEMBER 29, 1986

EXAMPLES OF FLEX COMMANDS

+++ list, pim ←

NUMBER EQU \$10

POPO EQU \$20

ORG \$100

START NOP

LDA *\$40

SUBA *NUMBER

STA DIFF

ADDA *POPO

STA DIFF+1

SWI

FCB 0

DIFF RMB 2

END

Which system are we in?

- An aside: Always watch the **"prompt"**
 - It indicates that the system expects input from the keyboard,
 - It tells you **which system** has control of the machine:
 - Monitor: **ROSY>**
 - FLEX: **+++**
 - Editor: **#**
 - Assembler:

Fig.1

Disks

- On a two-disk system, the *system disk* must be in *drive 0*
- To inspect what is on the disk 0, type:
 - + + + **FILES,0**
 - + + + **CAT,0**
 - + + + **DIR,0**
- End all commands with "Carriage Return"
- You may also type: *FILES 0* , or also *FILES 1* or *CAT,1*.
- To see how much space is left on Disk 1:
 - + + + **FREE 1**

RESULT OF "FILES"

FILES ON DRIVE NUMBER 0

ERRORS.SYS	FLEX.SYS	CAT.CMD	DIR.CMD	COPY.CMD
LIST.CMD	ASN.CMD	DELETE.CMD	RENAME.CMD	TTYSET.CMD
P.CMD	SAVE.CMD	EDIT.CMD	ASMB.CMD	ASM.CMD
APPEND.CMD	BUILD.CMD	EXEC.CMD	JUMP.CMD	DATE.CMD
O.CMD	LINK.CMD	VERSION.CMD	PROT.CMD	VERIFY.CMD
PRINT.CMD	QCHECK.CMD	I.CMD	XOUT.CMD	PUTLDR.CMD
HELP.CMD	TMODE.CMD	GETVAX.CMD	PRIVAX.CMD	FILES.CMD
DUMPDISK.CMD	FORMAT.CMD	LOAD.CMD	COPYF.CMD	CREATE.CMD
FILETEST.CMD	FREE.CMD	RUN.CMD	BASIC.CMD	MTEST09.CMD
EPROM.CMD	WPS.CMD	DELETE.HLP	BUILD.HLP	DATE.HLP
APPEND.HLP	ASN.HLP	FREE.HLP	GET.HLP	VERIFY.HLP
JUMP.HLP	CAT.HLP	COPYF.HLP	LIST.HLP	EXEC.HLP
COPY.HLP	I.HLP	P.HLP	PRINT.HLP	O.HLP
XOUT.HLP	RENAME.HLP	SAVE.HLP	QCHECK.HLP	WHELP.TXT
GUIDE.TXT	MEMO.TXT	MONCALLS.TXT	SINETAB.TXT	STARTUP.TXT
STARTUP2.TXT	DIVIDE.TXT	D_MULT.TXT	MOUSE.TXT	WELCOME.TXT
COPYD.TXT	SI.TXT	MOUSE.BIN	MOUSE.SYM	SI2.TXT
KERNEL.TXT				

RESULT OF "CAT"

CATALOG OF DRIVE NUMBER 0
DISK: FLTS0986 #1

NAME	TYPE	SIZE	PRT
ERRORS	.SYS	9	
FLEX	.SYS	23	
CAT	.CMD	3	
DIR	.CMD	5	
COPY	.CMD	5	
LIST	.CMD	3	
ASN	.CMD	1	
DELETE	.CMD	2	
RENAME	.CMD	1	
TTYSET	.CMD	2	
P	.CMD	1	
SAVE	.CMD	2	
EDIT	.CMD	28	
ASMB	.CMD	48	
ASM	.CMD	52	
APPEND	.CMD	3	
BUILD	.CMD	1	
EXEC	.CMD	1	
JUMP	.CMD	1	
DATE	.CMD	2	
-			
-			
-			
MOUSE	.TXT	15	
WELCOME	.TXT	2	
COPYD	.TXT	1	
SI	.TXT	1	
MOUSE	.BIN	2	
MOUSE	.SYM	2	
SI2	.TXT	1	
KERNEL	.TXT	34	

SECTORS LEFT - 708

Fig. 4

RESULT OF "DIR"

DIRECTORY OF DRIVE NUMBER 0
DISK: FLTS0986 #1 CREATED: 29-SEP-86

FILE#	NAME	TYPE	R	BEGIN	END	SIZE	DATE	PRT
1	ERRORS	.SYS	R	01-01	01-09	9	10-SEP-85	
2	FLEX	.SYS		01-0A	01-20	23	10-SEP-85	
3	CAT	.CMD		02-01	02-03	3	10-SEP-85	
4	DIR	.CMD		02-04	02-08	5	10-SEP-85	
5	COPY	.CMD		02-09	02-0D	5	10-SEP-85	
6	LIST	.CMD		02-0E	02-10	3	10-SEP-85	
7	ASN	.CMD		02-11	02-11	1	10-SEP-85	
8	DELETE	.CMD		02-12	02-13	2	10-SEP-85	
9	RENAME	.CMD		02-14	02-14	1	10-SEP-85	
10	TTYSET	.CMD		02-15	02-16	2	10-SEP-85	
				.				
				.				
				.				
				.				
				.				
				.				
				.				
				.				
				.				
				.				
80	WELCOME	.TXT		10-12	10-13	2	10-SEP-85	
81	COPYD	.TXT		10-1A	10-1A	1	17-SEP-85	
82	SI	.TXT		10-1B	10-1B	1	17-SEP-85	
83	MOUSE	.BIN		10-16	10-17	2	10-SEP-85	
84	MOUSE	.SYM		10-18	10-19	2	10-SEP-85	

Fig. 5

File Specification

- A complete *file specification* is:
 - **1.MYFILE.BIN**
 - **0.SNOOPY.TXT**
 - *drive.filename.extension*
- In many cases the 3 fields may *default* to *preset* values:
 - The *drive* number may *often* be *left out*, unless you want to force it;
 - The *filename* is *not needed* in **a few cases** (e.g assembler output files have the same name as the input file, unless you specify differently);
 - The *extension* may *often* be *left off*. (e.g. the editor expects Text file, the loader a Binary file.)

File Specification

- Maximum 8 characters in the filename: first must be a *letter*.
- **A-Z, 0-9, - and _** are allowed, **No others!**
- Extensions: **.SYS .CMD .BIN .TXT .OUT .SYM .HLP .BAK**

FLEX Commands

- The FLEX commands reside on disk (except MON and GET).
- + + + **0.LIST.CMD,1.SNOOPY.OUT**
- or shorter: + + + **LIST SNOOPY**
- Note: separation character is , or a space.
- The most often used FLEX commands are:
 - + + + **FILES** + + + **CAT**
 - + + + **DIR**
 - + + + **MON**
 - + + + **LIST,"file spec"[,"line range"][,N]**

FLEX Commands

- + + + **PRINT,"file spec"**
- + + + **COPY,"file spec1","file spec2"**
- + + + **SAVE,"file spec","begin addr","end addr"[,"tr. addr"]**
- + + + **DELETE,"file spec"**
- + + + **RENAME,"file spec1","file spec2"**
- + + + **P,"command"**
- + + + **O,"file spec","command"**
- + + + **GET,"file spec"**
- + + + **LOAD,"file spec"**
- + + + **ASM,"file spec" + options**
- + + + **EDIT,"file spec" + options**

FLEX Commands

- Examples of commands:
 - + + + **P,LIST,MYFILE.TXT**
 - + + + **O,CONTENTS,DIR,0**
 - + + + **LIST,CONTENTS**

Editor

- Evoked with: + + + **EDIT,"file spec"**
e.g. + + + EDIT,TEST1
- Two cases:
 1. File exists already. Editor reads it, then: #
 2. File does not exist. Editor responds with:
 - **NEW FILE**
 - **1.00=**
 - *Now you can start typing!*
- Our editor is **line oriented**, with the notion of "**current line**"

Editor

- Thus the command **P** will list the current line.
- You can force a line to become the current line:
 - **14P** will list line 14
 - **14P5** will list lines 14, 15, 16, 17, 18
 - **^P!** will list *all* lines
- Similarly:
 - **14D** will delete line 14,
 - **14D5** will delete lines 14–18 inclusive.
- **^** forces the current line to be at the **top** of the file,

Editor

- **!** forces the *last* line of the **range** to be at the **bottom** of the file.
- Don't try **^D!**
- Other important Editor commands:
 - **# RENUMBER** or **REN**
 - **# BOTTOM** or **B**
 - **# TOP** or **T**
 - **# LOG** or **L**
 - **# STOP** or **S**
 - **# FIND** or **F**
- **# FIND/string/** will find (and list) the line containing "*string*". For variants: see the manual.

EXAMPLE OF AN EDIT SESSION

```

+++edit,pim          ← EDIT COMMAND
DELETE BACKUP FILE (Y-N)? Y ← EDITOR ASKS QUESTION
*p!                  ← LIST WHOLE FILE

1.00=NUMBER EQU $10
2.00=POPD EQU $20
3.00= ORG $100
4.00=START NOP
5.00= LDA #$40
6.00= SUBA #NUMBER
7.00= STA DIFF
8.00= ADDA #POPD
9.00= STA DIFF+1
10.00= SWI
11.00= FCB 0
12.00=DIFF RMB 2
13.00= END

*01                  ← INSERT BEFORE LINE 1
.10= TTL FASULO      ← EDITOR TYPES .10 =
.20=*                ← USER TYPES THE LINE.
.30=*
.40=* This program has no useful purpose
.50=*
.60=*
.70=* It adds a few numbers, in immediate addressing mode
.80=* *
.90=*

1.00=*
1.10=*              ← TYPING * STOPS THE EDITOR
                    ← FROM ASKING MORE LINES.
SOME LINES RENUMBERED
1.00=*

*renumber           ← ANOTHER EDITOR COMMAND
*p!                 ← LIST ENTIRE FILE AGAIN

1.00= TTL FASULO
2.00=*
3.00=*

```

Fig. 6/7 begin

```

4.00=* This program has no useful purpose
5.00=*              COMMENTS ADDED
6.00=*
7.00=* It adds a few numbers, in immediate addressing mode
8.00=*
9.00=*
10.00=*
11.00=NUMBER EQU $10
12.00=POPD EQU $20
13.00= ORG $100
14.00=START NOP
15.00= LDA #$40
16.00= SUBA #NUMBER
17.00= STA DIFF
18.00= ADDA #POPD
19.00= STA DIFF+1
20.00= SWI
21.00= FCB 0
22.00=DIFF RMB 2
23.00= END
*7c/mde/mode/       ← MAKE CORRECTIONS
7.00=* It adds a few numbers, in immediate addressing mode
*11/c/ / define a few numbers
?                  ← EDITOR DOES NOT UNDERSTAND
*11c/ / define a few numbers/ ← AGAIN
?
*11c/$10/$10 define a few numbers/ ← NOW IT UNDERSTOOD
11.00=NUMBER EQU $10 define a few numbers
*13o                ← MODIFY LINE 13 WITH "OVERLAY"
13.00= ORG $100     ← ORIGINAL
OVERLAY            program starts at $100 ← OVERLAY
13.00= ORG $100     program starts at $100 ← RESULT
*14c/$40 here we start our job/
?                  ← EDITOR DOES NOT UNDERSTAND.
                    (STRING $40 DOES NOT EXIST IN LINE 14)
*14p                ← PROVE
14.00=START NOP
*14c/NOP/NOP here starts the job?
14.00=START NOP here starts the job? ← NOW IT IS OK

```

Fig. 6/7 continued

*14o
14.00=START NOP here starts the job? ← GET RID OF ?

OVERLAY

14.00=START NOP here starts the job.
*pl ← LIST FROM CURRENT LINE TO END
14.00=START NOP here starts the job.
15.00= LDA *\$40
16.00= SUBA *NUMBER
17.00= STA DIFF
18.00= ADDA *POPO
19.00= STA DIFF+1
20.00= SWI
21.00= FCB 0
22.00=DIFF RMB 2
23.00= END

ADD MORE COMMENTS
USING C AND O

*20c/SWI/SWI program must stop properly! ←
20.00= SWI program must stop properly!

*22o ←
22.00=DIFF RMB2

OVERLAY reserve two bytes for DIFF and DIIF+1
22.00=DIFF RMB2 reserve two bytes for DIFF and DIIF+1

*pl FINAL LISTING

1.00= TTL FASULO
2.00= *
3.00= *
4.00= * This program has no useful purpose
5.00= *
6.00= *
7.00= * It adds a few numbers, in immediate addressing mode
8.00= *
9.00= *
10.00= *
11.00=NUMBER EQU \$10 define a few numbers
12.00=POPO EQU \$20
13.00= ORG \$100 program starts at \$100

14.00=START NOP here starts the job
15.00= LDA *\$40
16.00= SUBA *NUMBER
17.00= STA DIFF
18.00= ADDA *POPO
19.00= STA DIFF+1
20.00= SWI program must stop properly!
21.00= FCB 0
22.00=DIFF RMB 2 reserve two bytes for DIFF and DIIF+1
23.00= END
* S ← QUIT THE EDITOR

Fig. 6/7 end

Fig. 6/7 continued

Editing of text

- To edit the text, the following commands are available:
 - **# APPEND/string/**
 - **# CHANGE/string1/string2/**
 - **# DELETE**
 - **# INSERT (below the current line)**
 - **# OVERLAY**
 - **# PRINT**
 - **# REPLACE (=DELETE + INSERT)**

Editing of text

- Examples:
 - **14P**
 - 14.00= HERE ADDA NUMBER
 - **# 14C/HERE/THERE/**
 - 14.00= THERE ADDA NUMBER
 - **#**
 - 14.10= * **This is an extra comment**
 - 14.20= * **and another.**
 - 14.30= **#**
 - **#**
- You **quit** the editor with **S**.

Assembler

- The assembler will be evoked with:
+++ **ASM,"file spec"** + *options*,
options... (see manual!)
- It may ask a few questions, and then the disk will start spinning.
- +++ **ASM,PROG** will use *PROG.TXT*.
- It will produce:
 - *Output listing* on the screen,
 - *Binary output* on disk: *PROG.BIN*,
 - *Symbol table* on disk: *PROG.SYM*

A SMALL EXAMPLE PROGRAM

BEFORE ASSEMBLY

```

NUMBER EQU $10
POPO EQU $20
ORG $100
START NOP
LDA *$40
SUBA *NUMBER
STA DIFF
ADDA *POPO
STA DIFF+1
SWI
FCB 0
DIFF RMB 2
END

```

AFTER ASSEMBLY

0010	NUMBER	EQU	\$10
0020	POPO	EQU	\$20
		ORG	\$100
0100	START	NOP	
0101 86 40		LDA	*\$40
0103 80 10		SUBA	*NUMBER
0105 B7 010F		STA	DIFF
0108 8B 20		ADDA	*POPO
010A B7 0110		STA	DIFF+1
010D 3F		SWI	
010E 00		FCB	0
010F	DIFF	RMB	2
		END	

0 ERROR(S) DETECTED

SYMBOL TABLE:

DIFF 010F NUMBER 0010 POPO 0020 START 0100

```

+++asm.plm          ← INVOKE ASSEMBLER
DELETE OLD BINARY (Y-N) ? y ← QUESTION
*                  ← OUTPUT LISTING STARTS
*
* This program has no useful purpose
*
* It adds a few numbers, in immediate
  addressing mode
*
*
*
0100 0010  NUMBER EQU $10  define a few numbers
0100 0020  POPO  EQU $20
0100 12    ORG  $100  program starts at $100
0101 86 40  START NOP    here starts the job
0103 80 10  LDA  *$40
0105 87 010F SUBA *NUMBER
0108 88 20  STA  DIFF
010A 87 0110 ADDA *POPO
010D 3F    STA  DIFF+1
010E 00    SWI    program must stop properly!
F+010F    FCB  0
          DIFF  RMB 2    reserve two bytes for DIFF
                      and DIFF
          END

0 ERROR(S) DETECTED ← NO ERRORS !

SYMBOL TABLE:      ← SYMBOL TABLE

DIFF 010F NUMBER 0010 POPO 0020 START 0100

```

Fig. 8

Assembler

- To get a *hardcopy* of the output listing:
 + + + **P,ASM,PROG**
 + + + **O,PROG.OUT,ASM,PROG**
 + + + **PRINT,PROG.OUT**
- Diagnostics messages are interspersed with the output listing. At the end of the listing you get:
47 ERROR(S) DETECTED
 followed by the symbol table.

Running the Program

- You load your program with:
+++ **LOAD,PROG.BIN** (or simply PROG)
- This will also load the **Symbol Table**. Thus ROSY knows and recognises the symbols you used. Then, in your **debugging**, you may use **Symbols**, instead of *addresses*!
- After loading you do:
+++ **MON**
- This will be immediately overwritten by: **ROSY>**

Running the Program

- Now you may:
 - Set breakpoints,
 - Inspect memory,
 - Modify memory,
 - Decode,
 - Set registers,
 - etc, etc.
- Usually you will do:
 - ROSY> **GO START** or
 - ROSY> **GO 2100** or
 - ROSY> **GO BEGIN**
- Only one thing now remains to be done:

```

+++ LOAD, PIM.BIN
+++ MON          ← FLEX COMMANDS "LOAD" AND "MON"
ROSY >          ← BACK IN ROSY
ROSY > decode 100 110 ← CHECK PROGRAM IS IN MEMORY
0100 12      NOP
0101 86 40   LDA  *$40
0103 80 10   SUBA *$10
0105 B7 010F STA  $010F
0108 8B 20   ADDA *$20
010A B7 0110 STA  $0110
010D 3F 00   MON  *$00
010F 30 50
ROSY > m-H 100 11F

                MEM LIST
ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
0100 12 86 40 80 10 B7 01 0F 8B 20 B7 01 10 3F 00 2A ..@.....?..0
0110 3C C8 A5 EC 4A 44 56 44 56 44 56 17 1F F2 ED 4A P...JDVDVDV.....J
ROSY > m-mo 10F
010F 2A 00 ← SET TWO BYTES TO 0
0110 3C 00
0111 C8
ROSY >
ROSY > go 100      RUN THE PROGRAM

ROSY > m-H 100 11F

                RESULTS ARE HERE I
ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
0100 12 86 40 80 10 B7 01 0F 8B 20 B7 01 10 3F 00 30 ..@.....?..0
0110 50 C8 A5 EC 4A 44 56 44 56 44 56 17 1F F2 ED 4A P...JDVDVDV.....J
ROSY > f          BACK TO FLEX

+++delete, pim.bin      DELETE A FILE

DELETE "O.PIM.BIN" ? y
ARE YOU SURE? y        IS THIS WHAT YOU WANTED ?
+++files,0             CHECK THAT FILE HAS BEEN DELETED

```

FILES ON DRIVE NUMBER 0

ERRORS.SYS	FLEX.SYS	CAT.CMD	DIR.CMD	COPY.CMD
LIST.CMD	ASN.CMD	DELETE.CMD	RENAME.CMD	TTYSET.CMD
P.CMD	SAVE.CMD	EDIT.CMD	ASMB.CMD	ASM.CMD
APPEND.CMD	BUILD.CMD	EXEC.CMD	JUMP.CMD	DATE.CMD
O.CMD	LINK.CMD	VERSION.CMD	PROT.CMD	VERIFY.CMD
PRINT.CMD	QCHECK.CMD	I.CMD	XOUT.CMD	PUTLDR.CMD
HELP.CMD	TMODE.CMD	GETVAX.CMD	PRIVAX.CMD	FILES.CMD
DUMPDISK.CMD	FORMAT.CMD	LOAD.CMD	COPYF.CMD	CREATE.CMD
FILETEST.CMD	FREE.CMD	RUN.CMD	BASIC.CMD	MTEST09.CMD
EPROM.CMD	WPS.CMD	DELETE.HLP	BUILD.HLP	DATE.HLP
APPEND.HLP	ASN.HLP	FREE.HLP	GET.HLP	VERIFY.HLP
JUMP.HLP	CAT.HLP	COPYF.HLP	LIST.HLP	EXEC.HLP
COPY.HLP	I.HLP	P.HLP	PRINT.HLP	O.HLP
XOUT.HLP	RENAME.HLP	SAVE.HLP	QCHECK.HLP	WHELP.TXT
GUIDE.TXT	MEMO.TXT	MONCALLS.TXT	SINETAB.TXT	STARTUP.TXT
STARTUP2.TXT	DIVIDE.TXT	D_MULT.TXT	MOUSE.TXT	WELCOME.TXT
COPYD.TXT	SI.TXT	MOUSE.BIN	MOUSE.SYM	SI2.TXT
KERNEL.TXT	PIM.TXT	PIM.BAK		
		PIM.BIN HAS GONE		
copy pim.txt,prime.txt				

Fig. 9 continued

Fig. 9

**KEEP YOUR
FINGERS
CROSSED**