COMPUTATION CENTER
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

The FORTRAN-FAP Monitor System

By F. J. Corbató

DESCRIPTION

The FORTRAN-FAP Monitor System (FMS) is a supervisory program that processes a sequence of jobs under the direction of the control cards included with each job. An FMS job deck consists of a collection of subprograms, and control cards, followed by data cards (if any) of the user. The processing options of each job are specified by the control cards in the job decks. Each of the subprograms may be supplied in three different forms: written in the FORTRAN source language, written in the FAP source language, or in the previously translated machine language form of relocatable binary (BSS form). The normal processing of a job consists of appropriately translating all subprograms written in a source language, writing all translated subprograms in BSS form on the punched-card output tape, obtaining from the FMS library tape in BSS form all subprograms not explicitly supplied, and transferring control to the beginning of the "main" subprogram in the job. The user's program, when finished, returns control to the Monitor so that the next job may be initiated. This is done by executing a CALL EXIT statement or by reading to exhaustion the data cards from the input tape.

It is possible to circumvent the dilemma which arises when a collection of subprograms and their COMMON data block in upper memory will not fit in the available core memory space. The program is judiciously subdivided and augmented into smaller groups called "chain links", such that each link is a complete set of subprograms, each link has a "main" subprogram, and each link and the COMMON data block fit in the available memory space. The Monitor, before starting execution of a job, records each link on tape and starts executing the first link. The user's first link, at an appropriate time in the calculation, uses a library subprogram and calls in another

link (not necessarily in the link input sequence), and so forth, until ultimately control is returned to the Monitor. Each time a link is brought in, it is in its original unexecuted state, and it destroys the previous link; thus all communication between links is done via the data stored in the upper memory COMMON block.

CONTROL CARDS

The Monitor Control Cards are always designated by an asterisk in column 1 followed by the control words in columns 7-72 (with blank columns ignored). Control cards cannot contain any text but the key words.

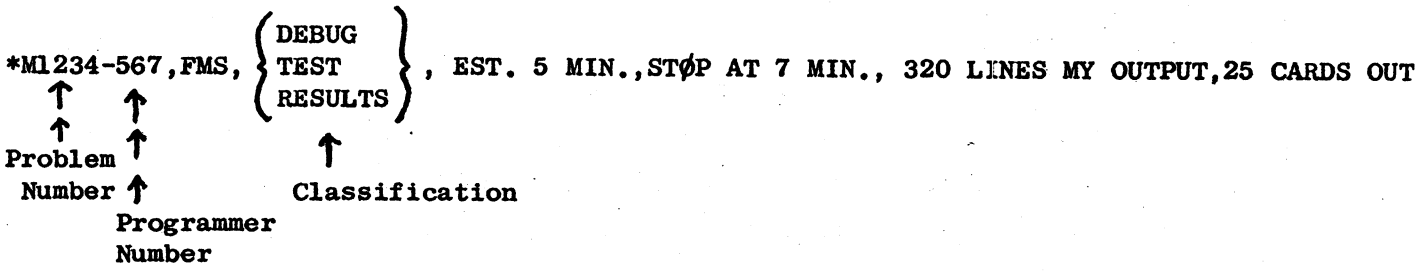
	Card	Position in Job	Function
General Control Cards			
1.	(ID)	First Card	Job identification (see general information section for format).
2.	XEQ	Must follow ID card and first subprogram or any other control card.	Instructs the Monitor to start execution of the job after all subprograms are translated provided no translation error diagnostics have occurred. If omitted, no execution occurs.
3.	DATA	Must follow the physically last subprogram of a job to be executed and precedes the data, if any, for that job.	Signals the Monitor that no more subprograms in this job follow on the input tape and initiates execution, if any.
4.	CHAIN(R,T)	Must precede the physically first subprogram of each chain link.	Identifies the chain link by record number, R,($0<R<32768$), and tape, T, where T is normally B_1 but may be B_2 or B_3.
5.	(Any non-control card text)	Anywhere after ID card and before DATA card except within a subprogram deck.	Arbitrary comment by user printed on-line or off-line.
FORTRAN Subprogram Translation Control Cards			
1.	FORTRAN	(Optional.) May precede the FORTRAN subprogram source cards.	A comment that the Monitor should translate to BSS form the following subprogram written in the FORTRAN language.
2.	LIST	Must precede FORTRAN source subprogram to which pertinent.	Causes FAP listing of object subprogram to appear in translation output.

	Card	Position in Job	Function
FAP Subprogram Translation Control Cards			
1.	FAP	Must precede FAP subprogram symbolic cards.	Instructs Monitor to translate the following subprogram written in the FAP language.
Other Control Cards Not Normally Used			
1.	LIBE	Must precede FORTRAN source subprogram to which pertinent. (Not pertinent to FAP.)	Instructs the Monitor to search the Monitor Library during translation for all subprograms called for by this subprogram and to include these with the object subprogram written on the off-line punch output tape.
2.	CARDS ROW	Must precede FAP or FORTRAN source subprogram to which pertinent.	Instructs the Monitor to punch on-line 704-standard relocatable row binary cards, preceded by a BSS loader if a main program. The subprogram is also written on the off-line punch tape.
3.	CARDS COLUMN	Must precede FAP or FORTRAN source subprogram to which pertinent.	Instructs the Monitor to punch on-line column binary relocatable cards (no loader) and not to write the subprogram on the off-line punch tape.
4.	END TAPE	Placed after a batch of jobs where each job is followed on the input tape by an end-of-file mark. (Usually inserted by the computer or peripheral operator.)	Signals the Monitor there are no more jobs on the input tape.
5.	(No asterisk in column 1 but instead a 7- and 8-row punch.)	Placed after each job only when writing the input tape from on-line. (This card is normally inserted between jobs by the computer operator.)	Signals Monitor card-to-tape simulator to write an end-of-file mark.

GENERAL INFORMATION

1. ID Card of the Computation Center

A permissible format is as follows (see Procedures Handbook for further details):



Blanks are ignored as well as any non-digits in the last four fields. The classification should be: DEBUG for a program being translated or not yet operating normally, TEST for a program still being tested for correctness of method, and RESULTS for a program producing useful answers. The Monitor of the Center will stop a program which attempts to produce output printing in excess of 125% of the estimate.

2. Optimum Sequence of Subprograms

To minimize tape movement time by the Monitor the optimum sequence of subprograms is 1) all FAP subprograms of job, 2) followed by all FORTRAN subprograms, 3) followed by all binary subprograms.

A sample FMS deck setup is:

```
*     (ID)
*     XEQ
*     FAP
      ...
      END
*     FORTRAN
*     LIST
      ...
      END
*     BINARY SUBPROGRAMS FOLLOW
      ...
*     DATA
      ...
```

3. FMS Input and Output Tapes

FMS input tapes are normally written off-line with an end-of-file mark after each job and an END TAPE control card at the end of each batch of jobs. The Monitor is started by an on-line "start" card, and if there are no further cards in the on-line card reader, the jobs on the input tape are processed in sequence.

If the Monitor is started and there are further cards in the on-line reader, the input tape is rewound, and written using an internal card-to-tape simulator. When input is from on-line, special cards (column 1: no asterisk but 7- and 8- row punches) are used after each job to signal the simulator to write end-of-file marks on the input tape.

The punched output tape of FMS should always be written in the column binary mode. Punching by the user should be preceded by a FLIP card (a FLIP card is visually readable containing 6 characters in stencil form; if read by the MIT version of the Monitor it appears as a column binary card of zero words and hence is ignored).

The Monitor, when processing the ID card, automatically writes 4 FLIP cards for the job consisting of: problem number, programmer number, date, and time. In addition, each translated subprogram is preceded by 3 FLIP cards consisting of date, time, and subprogram name. These 3 cards can be flipped together (so that the name is first) and be kept with the binary subprogram.

4. FORTRAN Subprogram Information

Each FORTRAN subprogram should terminate with an END statement without any arguments.

5. Library Subprograms Which Are Often Used With FMS

a) For convenience at the Center, the input-output subprograms in the FMS library, which are used by FORTRAN statements, have been modified so that the statements READ, PRINT, and PUNCH cause the corresponding off-line operation. The sense switches are all up normally and should not be used in programs. However, if the programmer is present when sense switch 3 is depressed, the on-line card reader is read instead of the input tape, when sense switch 4 is depressed the on-line punch is operated instead of writing the off-line punch output tape, and when sense switch 5 is depressed, the information being written on the off-line output tape is repeated on-line.

b) CHAIN

The last executed statement of a link that is to call a succeeding link for execution in a FORTRAN subprogram must be of the form,

CALL CHAIN (R,T)

where T is literally B1, B2 or B3.

Because of an unclean piece of coding in the Monitor, the corresponding FAP calling sequence does not directly follow. It should be

```
TSX     $CHAIN,4
PZE     RLØC
PZE     TLØC
```

where location RLØC contains PZE 0,,R and location TLØC contains PZE 0,,1 or PZE 0,,2 or PZE 0,,3 corresponding to tapes B1, B2 or B3.

Entry to CHAIN will cause the link, which at compilation time had been specified by the control card,

CHAIN (R,T)

to be read into memory and started at the "main" subprogram entry point.

c) (F2PM) and FTNPM

The standard FORTRAN-FAP System (FMS) has available two post-mortem subprograms DUMP and PDUMP for terminal and breakpoint dumps respectively. These subprograms are clumsy to use if dumping the contents of subprograms. Perhaps more seriously, each change in request for DUMP and PDUMP usually needs a new FORTRAN compilation (which requires at least a minute of computer time). To eliminate the above deficiencies as well as the absence of facilities for identifying text, different word formats, and tape dumps, a post-mortem subprogram (F2PM) has been prepared by the Center for FMS that executes either breakpoint requests or requests given on Hollerith control cards placed after the data cards of a run. Because (F2PM) is entirely in the user's program, it is faster in operation than the DUMP or PDUMP subprograms which must be brought in from the FMS system tape for each usage request.

The Center version of the FMS system tape is modified so that the subprogram storage map of a job at the end of loading is saved and stored at the end of all subprograms. The FORTRAN compiler is modified so that all FORTRAN main subprograms place a transfer to (F2PM) in location 7; (to anticipate any stops or infinite loops). Similarly, all FAP main subprograms should execute the instructions CLA $(F2PM), STØ 7. Programs should end by a CALL EXIT (which will automatically call (F2PM)). If a job reaches a CALL EXIT statement, the (F2PM) subprogram first examines the input tape; unread data cards are printed off-line. An automatic post-mortem section and any request cards that the user has placed at the end of his data cards are executed by (F2PM) and control is returned to the Monitor for the next job.

If the user's program should stop or loop endlessly, a manual entry to (F2PM) is started by the computer operator storing the contents of the MQ register in location 6, entering the Instruction Location Counter into the MQ register, and executing a TTR 7 instruction.

A break-point manner of using (F2PM) is for the programmer to include and execute in a FØRTRAN subprogram the statement

CALL FTNPM (arg1,arg2,...)

where each of the arbitrary number of arguments of the CALL is Hollerith text of the form $nHx_1x_2...x_n$ and corresponds to the punched-range from column 13 onward in an equivalent FORTRAN PM card. After (F2PM) has executed the Automatic and Request post-mortems, control is returned to the next statement after the CALL in the user's program.

Automatic Section of the Post-Mortem

This section consists of the following operations:

1. The status of the machine conditions at the time of a stop in the user's program is printed off-line.

2. A printout is produced off-line of the index-register block consisting of the first four locations before the subprogram entry of the subprogram which contains the stop of the user's run.

3. The twenty words surrounding the program stop are printed.

FORTRAN PM Cards

FORTRAN Post-Mortem Cards, all of which are identified by the words FORTRAN PM punched in columns 1-12 (columns 8, 11, 12 must be blank), are of the three types:

1. Comment Cards. Comment Cards, which may occur in any sequence among the post-mortem cards, have an asterisk in column 13 followed by arbitrary text. Exception: Comment Cards may not precede Override Cards.

2. Override Cards. Override Cards, which are optional, serve to suppress the different parts of the Automatic Section of the post-mortem.

3. Request Cards. Request Cards have from 1 to 4 variable fields starting in column 13 with commas separating the fields. The first blank column initiates a remark field which is printed on the output unit designated in the request.

Typical Request cards are:

1) FØRTRAN(1) PM(2) NAME(ENTIRE),ALL,ØFF,REV

2) FØRTRAN(1) PM(2) NAME(3,/20),FLØ.

These cause

1) The subprogram NAME to be dumped in its entirety in 4 different modes, off-line, and in reverse sequence and 2) the subprogram NAME to be dumped off-line in forward sequence, in floating-point form from locations 3_{10} to 20_8 relative to the origin of NAME.

Further details are given in the Procedures Handbook and Memo CC-167.

d) DUMP and PDUMP

These subprograms are not recommended (see (F2PM) post-mortem above).

The subprogram DUMP is used by the FØRTRAN statement:

CALL DUMP (A_1, B_1, F_1,...A_n, B_n, F_n)

where A and B are variable data names indicating limits of core storage to be dumped. Either A_1 or B_1 may represent upper or lower limits. F_1 is a fixed point number indicating

the format desired, where 0 is octal, 1 is floating-point, 2 is the decrement and sign as a decimal integer, and 3 is octal with instruction mnemonics.

The core dump is effected as specified and control is transferred to the Monitor to initiate the next job. If no arguments are given, all of core storage is dumped in octal - this feature should not be used normally. The last format indication may be omitted, in which case it will be assumed to be octal.

The subprogram PDUMP is used in a break-point manner anywhere in a FORTRAN source subprogram by CALL PDUMP (A_1, B_1, F_1...,A_n, B_n, F_n). The argument format for A, B, and F are the same as those given for the CALL DUMP statement.

The difference between PDUMP and DUMP is that, after PDUMP is executed, the machine is restored to its condition upon entry, and control is returned to the next executable statement. The memory dumps appear with any other off-line output from the job.

Restriction On The Use of PDUMP

Do not use the CALL PDUMP statement when there is a chain link on tape B2 to be executed subsequently. Tape B2 is used by the PDUMP program for intermediate storage of the contents of core storage where the part of PDUMP brought in from the systems tape is loaded.

6. Tapes Used by FMS

Logical	Physical	Function
1	A1	System
4	A2	Input (not rewound by Monitor)
2	A3	Off-line printed output (not rewound by Monitor)
8	A4	Translation intermediate
5	B1	Monitor intermediate (not needed unless execution is requested)
6	B2	Translation intermediate and used by PDUMP during execution
7	B3	Translation intermediate and Monitor intermediate
3	B4	Off-line punched output (not rewound by Monitor)

6. Tapes Used by FMS (con't)

Logical	Physical	Function
10	B5	FMS subprogram library (for jobs requiring the maximum number of tapes, a special version of FMS is available with the library far out on the system tape A1).
9	A5	Available to user

7. Stops

The user should never intentionally stop his program but should instead either execute the statement CALL EXIT or read his data cards to exhaustion from the input tape (which will cause the reading subprogram to transfer to EXIT).

The Monitor itself does not stop except when there is machine or tape malfunction. Normally the operators will inform you of this situation but for reference the stops are given here:

50_8 1-CS stop. Bad library tape or check sum. Depress the Start Key to try again.

417_8 Card-to-tape Simulator stops. Stops with HPR 7777_8 in storage register. Illegal card character code.

421_8 Stops with HPR 77777_8 in storage register. Redundancy error in writing tape. For the stop at either 417_8 or 421_8 to re-write the bad record, empty the card reader hopper, and feed out all cards in the card reader (followed by the remainder of the deck) in the hopper; ready the card reader, and depress the Start Key.

25_8 Stop at object time. Redundancy tape check while reading a chain link at object time. Five attempts have been made. Location 23_8 will have tape number in the address.

33_8 Stop while Monitor reading in object program from A4. Tried 4 times.

8. IBM Manuals

The above description represents all the information most users will want (and need) to know about FMS. The IBM 709/7090 FORTRAN Monitor Manual (C28-6065) is loosely written and contains much operational detail

normally not the concern of the user. There is also an IBM 709 Fortran Operations Manual (C28-6066-1) which describes in detail the intricacies of making Monitor system tapes. Copies of both manuals are in the Document Room of the Computation Center.