CC-167-6

COMPUTATION CENTER
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

March, 1963

TO: All Programmers

FROM: L. Korn and J. F. Corbató

SUBJECT: Description of a Post-Mortem Subprogram (F2PM) for Use with the Fortran-Fap Monitor System Version II

INTRODUCTION

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 subprograms. Perhaps more seriously each change in request for DUMP or PDUMP needs a new Fortran compilation. To eliminate the above deficiencies as well as the absence of facilities for identifying text, other word formats, and tape dumps, a post-mortem subprogram (F2PM) has been prepared for the FMS system which executes either breakpoint requests or requests given on Hollerith control cards placed after the data cards of a run.

USAGE OF (F2PM)

(F2PM) exists as a FAP-coded subprogram in the library of the MIT FMS System Tape. It also uses from the library the subprograms; KILLTR, (TSH), (IOH), (IOU), and EXIT. In addition, the subprogram MOVIE) is placed in core by the BSS loader. (The present system tape has a modified version of the Monitor BSS Control section so that the user's subprogram storage map is saved as the subprogram MOVIE) and stored at the end of all the user's subprograms.) The Fortran compiler is modified so that all Fortran main subprograms CALL .SETUP which plants a transfer to (F2PM) in a lower core location for manual post-mortems. Similarly, all FAP main subprograms should execute a CALL .SETUP which initializes core for (FPT), (F2PM), and (TIME) to anticipate any stops or infinite loops.

Programs should end by a CALL EXIT which will automatically call (F2PM). If the user's program reaches the CALL EXIT statement, an automatic post-mortem supplemented by any requests specified by the user on "F2PM Request Cards" is given by (F2PM) before control is returned to the monitor for the next job. If the entire data file for the run has not been read, remaining data cards will be skipped over except that the first 25 unread data cards are printed off-line. The computer operator can, at any time, manually force a simulation of a programmed CALL EXIT which will give the terminal post-mortem associated with the CALL EXIT. In addition the interval timer may also initialize this call if the job running time exceeds the maximum time stated on the ID card.

This operator or timer restart procedure relies on the presence of (F2PM) for a smooth transition to EXIT to return control to the Monitor and to terminate the job in case of a stop, programmed or manual. Thus, users who do not wish to include (F2PM) in their program because of lack of storage space or other reasons may include the DUMMY (F2PM) subprogram in the input deck. Then, the regular (F2PM) is never brought in from the library, and no user's storage map is retained in core memory.

Dummy (F2PM), if called by the timer or by the operator via the manual restart procedure, sends control to EXITM, an entry point of the EXIT subprogram, which terminates the job. The Dummy (F2PM) will also bypass any breakpoint F2PM statements that were compiled into the user's program. This Dummy (F2PM) deck is available from the Dispatcher or the Open-File of FMS routines in Room 26-058.

For programmers who wish to save about 1000$_8$ locations and still wish to use the post-mortem there has been a tapeless (F2PM) written. This version of (F2PM) will do everything that the library tape version does, except process tape requests. A copy of this deck may be obtained in the Open-files in Room 26-058.

AUTOMATIC SECTION OF THE POST-MORTEM

This section consists of the following automatic 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. In addition, twenty words around the last executed instruction are also printed.

F2PM REQUEST CARDS

Basically there are three types of F2PM Request Cards, all of which are identified by the words Fortran PM punched in Columns 1-12 (columns 8, and 11, and 12 must be blank). The three types are:

1. Comment Cards. Comment cards, which may occur in any sequence among the request cards, have an asterisk in column 13 followed by arbitrary text. The text is printed off-line as soon as the card is read. Restriction: Comment Cards cannot precede override cards.

2. Override Cards. Override Cards, which are optional, serve to suppress the different parts of the Automatic Section which is executed whenever the first Request Card is read. Thus all Override Cards must precede the first Request Card The various overrides are obtained by starting with the minus sign (11-row punch) in column 13. The first blank column initiates a remark field which is printed off-line in the same way as the text of the Comment Card.

The phrase - (AUTOMATIC) suppresses the entire Automatic Section of the post-mortem. Similarly, -(LIGHTS) and -(STOP) suppress, respectively, the machine conditions and the index register block and words around the program. All overrides are reset each time (F2PM) is entered; they do not carry over between breakpoint or manual entries.

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

   a. The first variable field may have the following forms:

      1) NAME for the index register block post-mortem of the subprogram NAME. The main program is designated by (MAIN).
      2) NAME (ENTIRE) for a post-mortem of the entire subprogram NAME. Note that this does not include COMMON storage.
      3) NAME (R1, R2) for a post-mortem of the inclusive range from the relative address R1 to the relative address R2 in the subprogram NAME. (More precisely, as with all subsequent range numbers, the range is from the lesser of R1 and R2 to the greater of R1 and R2). The representations for R1, R2 and all subsequently-referred-to-numerical quantities are taken to be decimal unless they are preceded by a / in which case they are taken to be octal.
      4) (A1, A2) for a post-mortem of the inclusive range from the absolute address A1 to the absolute address A2.
      5) (TAPE) (T, F, TMODE, R, (WD1, WD2)) for a post-mortem of Tape T, file F, record R recorded in TMODE over the inclusive range from word WD1 to word WD2. TMODE should be BIN for binary records and BCD for binary-coded decimal records. Note that BCD tapes should generally be printed not dumped. If F is omitted, it will be assumed that file zero is specified. This is the only permissable deviation from the above format. By definition, a rewound tape is considered to be set for reading of file zero, record zero, word zero. No second, third or fourth field used with (TAPE) requests; all output is produced as though these fields were specified as DEC, FWD. Note that BCD data written in the binary mode will be garbled in the resulting dump. All tapes must have been written by a Fortran WRITE TAPE statement. Programmers are cautioned about using breakpoint tape post-mortems since the tape positions will be altered by the tape requests. Between breakpoint tape PM's, the PMed tape is left positioned at the end of the last physical record read. A logical record may consist of n physical records.) If the number of words requested exceeds the total number of words on the record, the PM is stopped at the end of record. All data channel traps are disabled. Since no Fortran binary record can exceed 255 decimal words plus a control word, PMs are printed in blocks of 256 or less words. The locations are printed relative to the beginning of the record requested. Word zero always contains the Fortran control word.

b. The second variable field contains the output mode for the contents of the locations in the range specified in the first variable field. The mode designations has the following forms:

1) ALL for instruction code, octal word, Fortran fixed-point integer, floating-point number, and BCD word. Output format in 2 columns.
2) OCT for instruction code and octal word. Output format in 4 columns.
3) FIX for Fortran fixed-point integers, (sign and decrement represented decimally). Output format in 6 columns. (Note: this is not the integer form used with MAD).
4) FLO for floating-point numbers. Output format in 5 columns.
5) DEC for MAD fixed-point integers and mixed modes. If the address, tag, and last two bits of the prefix equal zero, the word is printed as a Fortran fixed point integer. If tag, decrement, and last two bits equal zero, the words are printed in the octal mode. Output format is 5 columns. The DEC mode is used for dumping tapes.
6) BCD for BCD words. Output format in 6 columns. The above output modes contain relative and/or absolute octal locations, whenever pertinent. The consecutive locations of an output request range are placed sequentially in a column, with consecutive columns occurring from left-to-right. The printing of consecutive locations containing identical information is condensed by the phrase REPEAT appearing in the output.

c. The third variable field contains the sequence for consecutively listing the locations of the request range.

1) FWD designates the consecutive locations appearing in ascending sequence.
2) REV designates the consecutive locations appearing in descending sequence.

If the third field is omitted, FWD is assumed.
If the second and third fields are omitted, ALL, FWD is assumed.
The second and third fields may be omitted only in the combinations stated above. The terminal character should be blank when fields are omitted. At least one blank must separate the remark from the request.

EXAMPLES: NAME, FIX REMARK

NAME REMARK

The complete request card is printed preceding the output of each request.

BREAKPOINT REQUESTS

The user may compile requests into his program which will cause a breakpoint post-mortem each time that section of the program is executed. Thus, he may get a dynamic picture of certain arrays.

If the number of lines of programmer output or the maximum running time is exceeded during a breakpoint post mortem, the request is completed, and control is sent to EXIT. Thus the Post Mortem request cards are always interpreted.

1. Argument Breakpoint

Fortran Calling Sequence:

```
CALL FTMBP  (A(n), A(m))
CALL FTMBP  (A(n),A(m),3HXXX)
CALL FTMBP  (A(n),A(m),3HXXX,3HFWD)
```

Arrays to be dumped may be requested symbolically. In the first example, the registers encompassing A(n) and A(m) will be dumped in reverse order in the "ALL" mode. Deviations from the normal mode may be requested through a third and fourth parameter to be specified in BCD. The mode is specified by the words ALL, OCT, FIX, FLO, DEC, and BCD in the third parameter. To dump in the forward mode FWD is specified in the fourth parameter. The third and fourth parameters may be omitted, but the fourth may not appear without the third.

FAP Calling Sequence:

```
TSX   $FTMBP,4        TSX   $FTMBP,4        TSX   $FTMBP,4
PZE   A         or    PZE   A         or    PZE   A
PZE   B               PZE   B               PZE   B
                      PZE   C               PZE   C
                                            PZE   D
```

The area to be dumped encompasses A and B. If the third and fourth parameters are omitted, the post-mortem is given in the "ALL" mode in reverse. Mode can be specified as a third parameter. The modes are ALL, OCT, FIX, FLO, DEC, AND BCD. One of these three letter codes must appear at location C in the binary-coded decimal as in the above example, as

```
C     BCI     1,FIX
```

fixed point mode.

If the post-mortem is to be in the forward mode, a fourth parameter must be specified as FWD in binary-coded decimal, as

```
D     BCI     1,FWD
```

The Automatic section cannot be suppressed with Symbolic Argument breakpoint.

2. Hollerith Argument Breakpoint

Fortran Calling Sequence:

CALL FTNPM (Arg_1, Arg_2,.....,Arg_n)

where each of the arbitrary number of arguments of the CALL FTNPM is Hollerith text of the form $nHx_1x_2...x_n$ and corresponds to the punched-range from column 13 onward in the equivalent Fortran PM card.

This type of breakpoint is used primarily to request a postmortem of a subprogram other than the one in which the request appears. After (F2PM) has executed the Automatic and Request card post-mortems, control is returned to the next statement after the CALL in the user's program.

EXAMPLE:

```
CALL FTNPM(4OH(MAIN)(ENTIRE),OCT
This is a Fortran PM., 14HPRINT(ENTIRE) )
```

The above example would post-mortem the entire main subprogram and the entire subprogram PRINT.

USE OF (F2PM) WITH MAD

To perform a breakpoint request the user should EXECUTE FTNBP. (A(n),A(m)). The subprogram .SETUP, which is precompiled into each main program will initialise core for (FPT) and (F2PM). If the users program is terminated by an EXECUTE EXIT., all request cards will be performed.

Terminal PM and FTNBP requests may be used with MAD the same as with Fortran except that the FIX mode will not correctly interpret integers. Instead, the DEC mode should be used.

RESTRICTIONS:

Core locations 77461 (octal) to 77777 (octal) should be dumped in the octal mode only. The other modes use (IOH) which used that area for its temporary storage. The sequence of requests should be -(AUTOMATIC) or -(STOP) followed by (/77461, /77777), OCT followed by any other requests.

Since (F2PM) itself is constantly changing its own core locations, it therefore should never be included in any dump request.

Logical tape 4 must be the tape containing the (F2PM) request cards.

Programmers using (F2PM) must have their (IOU) table setup such that logical tape 4 corresponds to A2 and logical tape 2 corresponds to A3. This correspondence is normal at the Computation Center.