



DECUS

PROGRAM LIBRARY

DECUS NO.	FOCAL8-8
TITLE	MAGTAPE FOCAL
AUTHOR	John C. Alderman, Jr.
COMPANY	Georgia Institute of Technology Atlanta, Georgia
DATE	July 17, 1969
SOURCE LANGUAGE	FOCAL

MAGTAPE FOCAL

DECUS Program Library Write-up

DECUS No. FOCAL8-8

ABSTRACT

A sophisticated handler package for the TC58 IBM compatible magtape controller is available in the FOCAL language. Data is transmitted to/from the FNEW arrayed storage in FIELD 1. The features of I/O overlap with program execution, defeatable error diagnostics, and programmable tape density and unit selection have been incorporated into the coding. Syntax is via the LIBRARY command, and there is extensive error checking of user calls.

The FIELD 1 resident portion of the magtape handler is removable by other users, and the result is a general purpose LIBRARY command handler package, capable of being linked to any device using the interrupt. A functional argument is transmitted to the FIELD 1 coding, and any number of numerical arguments may be evaluated.

Part of the data acquisition system at the Nuclear Research Center at Georgia Tech is an 8K PDP8/I, with a TC58 IBM compatible tape controller and a TU20 7-track tape unit. The difficulties in programming for the TC58 controller are sufficient to intimidate many users. Therefore, a handler for the TC58, which uses the LIBRARY structure of the FOCAL language, has been implemented. This handler facilitates the use of all of the normal operations of the tape unit, without the difficulty of programming it in assembly language.

DATA STORAGE

In order to allow greater variable storage in a PDP-8 with 8K of core, a package has been written using the FNEW function call of FOCAL to store data in FIELD 1 (Ref. 1). The data is addressed as a one-dimensional array, and the syntax for filling an address of the array is: FNEW (I,X) where I and X are any evaluable expressions. The value of I (range $0 \leq I \leq 1224$) is taken as the address, and the value of X is stored there (and also returned as the value of the whole function). In order to read-out the contents of the array, the syntax is: FNEW (I), where the function is evaluated as the current contents of address I.

The Magtape handler also addresses values in this array, and thus data may be transferred to or from the array from either the running program or the tape controller. The handler checks address boundaries for accesses from both the program and the controller. In order to prevent a program from addressing the same FNEW addresses as the tape, the handler inhibits all programmatic FNEW addressing congruent with tape operations currently in progress. Upon completion of the tape operation, the program continues.

The Magtape handler is presently available in two dialects. The version most popular with users who take data to 36-bit computers for further processing is one that reads or writes records in 36-bit PDP-8 floating point format (Ref. 2). Another version handles 12 bit PDP-8 integers and is useful for other specific applications.

OTHER FEATURES

Errors concerning the magtape unit are of two general types: 1. Programmatic blunders, which are always flagged by an appropriate error diagnostic message (see Table 1). 2. Magtape operation errors which are normally flagged by the 722 6 diagnostic message. These errors may be determined by evaluating the FADC function, which has been preempted for the purpose (see Table 3).

USAGE

It is assumed that the user has considerable familiarity with the IBM compatible tape-record format. The TC58 maintenance manual (Chapter 1) has an excellent discussion of the subject. It is also assumed that the user has some experience with the FOCAL language.

Syntactical errors in calls to the tape handler will be flagged by their own diagnostics (see

Table 1). Tape controller errors (End-of-File, Parity Error, Record Length Incorrect, etc.) generate the magtape diagnostic, and the FADC function may then be used to evaluate the status of the tape unit after the error. The FADC function is evaluated by obtaining the status word from the controller (MTRS), masking it to suppress irrelevant status bits, and shifting the result right once to insure that the results are always a positive integer (see Table 3). For example, the End-Of-File mark may be sensed either during a SPACE, BACKSPACE, or GET operation. The FADC function would return the value of 32 if the EOF were the only error detected. Thus it is possible to program End-Of-File searches by using the SPACE or BACKSPACE commands.

One of the important parameters of input (GET) commands to the tape handler is the number of data points in the tape record. If the user specifies this incorrectly, an error will be detected. The handler also checks the range of the requested data transfer; and if any addresses are outside of the FNEW array, an error diagnostic message is given, and the transfer is not permitted.

LIBRARY COMMAND

The library (or L) command is used to direct the tape handler. The syntax for addressing the tape unit is LIBRARY FUNCTION, ARG1, ARG2. In some cases the arguments may be left out, and in all cases extraneous spaces and arguments will be ignored. The arguments mentioned may be (as usual) any evaluable expression. The Library command syntax requires the L character and a space as a terminator for the end of the first word. Thus the following are equivalent Library calls:

```
LIBRARY REWIND
LIBRARY R
L REWIND
L R.
```

Notice the space after the command. Also note that both the command and function names may be abbreviated to their first letter. The command allows termination of the command string by a carriage return (␣) or a semi-colon (;). In the latter case, any executable command may follow the semi-colon, including more LIBRARY commands.

The FUNCTION mentioned above is interpreted to be instructions for the tape controller, such as a rewind-to-beginning-of-tape, or space forward, space backward, write-data-onto-tape, get-data-from-tape, write-end-of-file, or several other actions. The first letter of the function word is decoded to initiate the appropriate action and set up the appropriate error conditions.

Functions Available

1. REWIND (or R). This causes the tape to return to the beginning-of-the-tape (BOT, or LOAD POINT). If the tape is already rewinding, or at BOT, this operation is ignored.

2. CLEAR (or C). When a tape "runs away" (as can happen when reading tapes), this function stops the tape unit and forces a rewind operation to the tape drive.

3. SPACE, ARG1 (or S, ARG1). Causes the tape unit to space forward ARG1 records. If any errors are detected, an error diagnostic is possible. (See note below.)

4. BACKSPACE, ARG1 (or B, ARG1). Similar to above but spaces in reverse direction.

5. EOF (or E). Writes an End-of-File record on the tape.

6. WRITE, ARG1, ARG2, (or W, ARG1, ARG2). Writes ARG1 36-bit (12-bit) PDP-8 floating point numbers on the tape from FNEW array starting at address ARG2.

7. GET, ARG1, ARG2 (or G, ARG1, ARG2). Reads ARG1 36-bit (12-bit) PDP-8 floating point numbers into FNEW array, starting at address ARG2.

8. INITIALIZE, ARG1 (or I, ARG1). The programmer may select unit number, tape density and type, interrecord gap length, and possible error diagnostic inhibition (see note below) by use of the initialization function. The argument is evaluated, converted to a 12 bit integer, masked to eliminate certain illegal combinations, and stored for loading into the command register of the TC58, along with the command code, at the next tape controller command. (Reference Table 2 and Pages 179, 180 of the Small Computer Handbook, 1968 Edition.) The handler reads and writes an odd parity only. The initialization word must be the sum of the tape density, unit number inhibition, and interrecord gap constants (see Table 2). The assembled value of 0 selects tape unit 0, 200 BPI, no diagnostic inhibition, and normal interrecord gap.

Note

Bit 9 (the "4" bit of ARG1) is used as a switch to inhibit the tape error diagnostic in case an error is detected during a tape operation. If this switch is set, it is incumbent upon the user to check the status of the tape operation via the FADC function before proceeding with more tape operations. This facility has been included to allow users to search for end-of-file records during spacing. Thus a spacing operation encountering an end-of-file, with the switch set, will not give an error diagnostic but will stop the tape and leave the error code in the register reserved for the FADC function.

Examples

1. Rewind the tape.

```
*LIBRARY REWIND
```

2. Rewind, write locations 10 through 20, re-wind, read the numbers back into 100 through 110.

```
*L REWIND:L WRITE,10,20;LR;L GET, 10, 100  
*
```

3. Rewind, then write contents of 0 through 200 in 10 word length records.

```
*LR;FOR I=0,10,190; L W, 10,I
```

4. Write End-of-File, rewind, space forward to end of file, back space over it, type error value.

```
*L E;L R;L S,100 (We know that only 20 records  
are on tape before EOF.)  
?22.<6 (found the EOF)  
*L B,1  
?22.<6 (found it in backspacing)  
*T FADC()  
32 (EOF still in storage as  
error)
```

5. Do same as before, but count number of records to EOF.

```
*L E;L R;F I=1, 100; L S,1  
?22.<6  
*T FADC(), " ",I, !  
32 20
```

6. Clear the tape controller, disable error flag, and space to first EOF.

```
3.1 I (FADC()-32)3.2,3.3,3.4  
3.2 I (FADC())3.4,3.1,3.4  
3.3 R  
3.4 T "ERROR CODE", %4.0,FADC(), !;QUIT  
*L C;L 1,4;L S,1000; D 3
```

7. Find the maximum and minimum of the next N records, length M.

```
1.1S MI=1E200;S MA=0;F I=1,N;L G,M,1;F J=1,M;  
D 2  
1.2Q  
2.1 IF (MAX-FNEW(J))2.2;S MAX FNEW(J)  
2.2 IF (FNEW(J)-MIN)2.3;S MIN FNEW(J)  
2.3 R
```

LIBRARY Command Decoding Package

Many FOCAL users have successfully added patches to the language implementing special functions (See FOCAL - How to Write New Subroutines and Use Internal Routines, D. E. Wrege, Proceedings Spring, 1969, DECUS Conference.) The language has available an unimplemented command, which may be accessed by the "L" or LIBRARY command. The tape handler is a successful implementation of the L command, and much of the utility functions

of the handler may be used for other purposes. All of the calls to FOCAL subroutines are contained in patches within FIELD 0 and form a working set of utility functions that the user may link to coding in FIELD 1.

A listing of MAGTAPE FOCAL is available from the author (in limited numbers) for those who desire to add such a package to their own installations. A brief description of the utility functions used in MAGTAPE FOCAL follows:

1. LIBRARY command recognition: This routine branches to FIELD 1 with the accumulator containing the ASCII code of the first character after the space following the L command. Presumably the user will do a SORTJ in FIELD 1 to decode the actual operation desired.

2. Argument Evaluator: This subroutine (ARGUMENT) is called from FIELD 1 to evaluate arguments preceded by a comma, which are then available to the FIELD 1 coding. Any number of arguments may be evaluated in this manner.

3. FNEW Function (See Ref. 1) which allows the program to store/retrieve data from the area of FIELD 1 assigned to the purpose.

4. CLEANUP routine, for skipping extraneous arguments to a semi-colon or return, and then giving control to the FOCAL executive.

5. An interrupt link to FIELD 1 for servicing additional interrupts.

6. A FIELD 1 routine to produce a normal FOCAL error diagnostic.

7. The FADC function which may be used to evaluate a status word stored in FIELD 1.

REFERENCES

Ref. 1 "Storage of Data Arrays in FOCAL," D. E. Wrege, -submitted to DECUS Library for publication. Also see "Focal - How to Write New Subroutines and Use Internal Routines," D. E. Wrege - proceedings spring, 1969, DECUS Conference.

Ref. 2 "PDP-8 Floating-Point System Programming Manual" (Dec, 1965)

TABLE 1
MT FOCAL DIAGNOSTICS

?22. 6	Tape control error, use FADC()
?23.;3	Error in argument of Library call
?25.:9	FNEW address too large
?24.06	Unknown Library function
?35.72	Initial Library address too large
?35.75	Final Library address too large
?35.97	Negative Library argument

TABLE 2
INITIALIZATION FACTORS FOR TC-58 CONTROLLER

Factor	Value (Decimal)
200 BPI	0
556 BPI	1
800 BPI	2
800 BPI/9 Track	3
3 Inch I.R. Gap	64
Unit 0	0
Unit 1	512
Unit 2	1024
Unit 3	1536
Unit 4	2058
Unit 5	2560
Unit 6	3072
Unit 7	3584

TABLE 3
TC-58 CONTROLLER STATUS AS READ BY FADC FUNCTION

Code	Error Condition
1	Bad tape (2 characters all zeros in middle of record)
2	Data Request Late (Data break too late)
4	Record length incorrect
16	End of tape (end point) reached
32	End-of-File detected
64	Parity error (lateral or longitudinal)
256	Beginning of tape (load point) reached

MAGTAPE FOCAL
MTF-37

```

FIELD 0
/TAPE HANDLING ROUTINES FOR FOCAL W.
/USES HOLES IN FPP.
/      5571-5577
/      6171-6177
/      6571-6576
/      5755-5773
/      7177176
/
/DEFINITIONS
MTSF=6703
MTCR=6711
MTTR=6721
MTAF=6712
MTRC=6724
MTLC=6716
MTRS=6706
MTGO=6722
INTEGER=52
FLAC=44
SPNOR=4521
GETC=4506
CHAR=142
PUSHJ=4501
POPJ=5502
EVAL=1603
ERROR4=4526
EFUN3I=100
FNTABF=376
PUSHA=4503
POPA=1413
SORTC=4511
COMGO=1171
GLIST=1406
FNEWTEST=0
LIMIT=2334
/
/SETUP FOR FADC() TO FETCH MT STATUS
/
*404
0404 7571      XADC
/
/SET UP FOR UPPER FIELD SORTJ
/COMMAND DECODER
*COMGO+10
1201 5571      TAPE
*5571
5571 4521      TAPE,  SPNOR
5572 1142      TAD CHAR
5573 4503      PUSHA
5574 5775      JMP I T1
5575 6171      T1,   XT1
5576 4526      MTFE,  ERROR4  /MAG-TAPE-FATAL-ERROR-?22.<6
*6171
6171 4506      XT1,   GETC
6172 4511      SORTC
6173 1405      GLIST-1      /SPACE,COMMA,SEMICOLON,CR LEGAL
6174 7410      SKP
6175 5371      JMP .-4 /TERMINATOR NOT FOUND
6176 5777      JMP I T2

```

```

6177 6571 T2,      XT2
          *6571
6571 4521 XT2,      SPNOR  /MOVE PAST TERMINATOR
6572 1413      POPA   /GET COMMAND CHARACTER
6573 6213      CDF CIF 10
6574 5775      JMP I  .+1
6575 7225      SORT1J
          /
          /
          /GETS LIBRARY ARGUMENT:CALLS ERROR4 IF NO COMMA
          /RETURNS WITH LOW ORDER INTEGER PART OF ARGUMENT
          /
          *5755
5755 0000 ARGUME, 0
5756 1142      TAD CHAR
5757 1371      TAD MCOMMA
5760 7640      SZA CLA
5761 4526      ERROR4 /NO COMMA-?23.;3
5762 4506      GETC   /MOVE PAST COMMA
5763 4521      SPNOR
5764 4501      PUSHJ
5765 1603      EVAL   /EVALUATE ARGUMENT
5766 4452      JMS I INTEGER
5767 6213      CDF CIF 10
5770 5755      JMP I ARGUMENT
5771 7524 MCOMMA, -254
          /
          /
          /IGNORES ADDITIONAL CHARACTERS TO ; OR CR,
          /AND THEN EXECUTES COMMAND IN ACC AT CALL
          /
5772 3375 CLEANU, DCA .+3 /ENTER WITH ACC= EXIT INSTRUCTION
5773 4511      SORTC
5774 1407      TERM-1
5775 7402      HLT
5776 4506      GETC   /GET ANOTHER CHARACTER
5777 5373      JMP  .-4
          /
          /FOCAL OVERLAY TO ALLOW STORAGE OF DATA ARRAYS
          /FOCAL W
          /CALL: FNEW(J,Z) - STORES Z IN APRAY ELEMENT J
          /      3 WORD FLOATING POINT VARIABLES
          /      FNEW(J) GETS ARRAYED DATA J
          /
          /LIMITS:0<J<1237
          /FNTABF=376
          /
          *FNTABF+15
0413 7346      XFNEW
          *7346
7346 4452 XFNEW,  JMS I INTEGER  /MAKE 12 BITS
7347 1373      TAD MOST
7350 7700      SMA CLA
7351 4526      ERROR4 /TOO LARGE FNEW ADDRESS - ?25.;9
7352 6212      CIF 10
7353 5774      JMP I MOST+1  /TEST RANGE OF J
7354 1046      TAD 46
7355 7104      RAL CLL
7356 1046      TAD 46
7357 4503      PUSHA
7360 4521      SPNOR
7361 1142      TAD CHAR
7362 1375      TAD MCOM
7363 7640      SZA CLA
7364 5776      JMP I XFGET
7365 4506      GETC

```



```

7366 4501          PUSHJ
7367 1603          EVAL
7370 1413          POPA
7371 5772          JMP I .+1
7372 7154          XFNEW2
7373 5444 MOST,    -LIMIT /PROTECTS CODING IN UPPER CORE
7374 7314          RANGETEST
7375 7524 MCOM,    -254
7376 7554 XFGET,   XFNEW3
/
*7154
7154 3163 XFNEW2,  DCA PNTR
7155 1044          TAD FLAC
7156 6211          CDF 10
7157 3563          DCA I PNTR
7160 2163          ISZ PNTR
7161 1045          TAD FLAC+1
7162 3563          DCA I PNTR
7163 2163          ISZ PNTR
7164 1046          TAD FLAC+2
7165 3563          DCA I PNTR
7166 6201          CDF
7167 5500          JMP I EFUN3I
/
*7554
7554 1413 XFNEW3,  POPA
7555 3163          DCA PTR
7556 6211          CDF 10
7557 1563          TAD I PTR
7560 3044          DCA FLAC
7561 2163          ISZ PTR
7562 1563          TAD I PTR
7563 3045          DCA FLAC+1
7564 2163          ISZ PTR
7565 1563          TAD I PTR
7566 3046          DCA FLAC+2
7567 6201          CDF
7570 5500          JMP I EFUN3I
7571 1373 XADC,    TAD .+2
7572 5355          JMP XFNEW3+1
7573 7574          COMBED-1
/
/LINK FOR INTERRUPT HANDLER FOR MAG TAPE
/
*2646
HINBUFF=37
2646 6011          RSF
2647 5252          JMP .+3
2650 6012          RRE
2651 3037          DCA HINBUFF
2652 6213          CIF CDF 10
2653 4562          JMS I XMTINT
2654 6244          RMF
*162
0162 7550 XMTINT,  MTINT
PTR,
0163 0000 PNTF,    0
0164 0615 XPFOC,  615 /PFOC
0165 2727 XROME,  2727 /POINTER TO ERROR SUBR.
/
*7225
/
FIELD 1 /COMMAND DECODEF FOR TAPE HANDLER
/CALLED BY TAPE IN LOWER CORE
/
/

```

```

7225 7041 SORTIJ, CIA /FIELD 1 SOFTJ
7226 3360 DCA COML
7227 1303 TAD COMLST /GET ADDRESS OF COMMANLIST
7230 3361 DCA TEMP
7231 1761 TAD I TEMP /GET LETTFE
7232 7510 SPA /LIST IS ENDED BY NEGATIVE NO.
7233 4252 JMS ERR /NOT IN LIST ?29.27
7234 1360 TAD COML
7235 2361 ISZ TEMP
7236 7640 SZA CLA /MATCH?
7237 5231 JMP *-6 /NO
7240 1303 TAD COMLST
7241 7041 CIA
7242 1361 TAD TEMP /YES: COMPUTE ADDRESS
7243 3671 DCA I XCOMM /LOAD COMMAND
7244 1361 TAD TEMP
7245 1272 TAD DIFF
7246 3361 DCA TEMP
7247 1761 TAD I TEMP /GET ADPCESS
7250 3361 DCA TEMP
7251 5761 JMP I TEMP /DO DOUBLE INDIRECT
7252 0000 ERR, 0
7253 7300 CLA CLL
7254 1252 TAD ERR
7255 6203 CIF CDF
7256 3662 DCA I DIAG
7257 1270 TAD XERFOR
7260 5661 JMP I .+1
7261 5772 XCLEAN, CLEANUP
7262 2726 DIAG, 2726
7263 7300 EXIT, CLA CLL
7264 1267 TAD OKEXIT
7265 6203 CDF CIF
7266 5661 JMP I XCLEAN
7267 5564 OKEXIT, JMP I XPROC
7270 5565 XERFOR, JMP I XBOMB /TO 2727, FIELD 0
7271 7433 XCOMM, COMM
7272 0054 DIFF, COMG01-COMLIST-1
/
/
COMLIST=.
7273 0322 322 /R(EWIND)
7274 0307 307 /G(ET)
7275 0311 311 /I(NITIALIZE)
7276 0327 327 /W(RITE)
7277 0305 305 /E(OFF)
7300 0323 323 /S(PACE)
7301 0302 302 /B(ACKSPACE)
7302 0303 303 /C(LEAP)
7303 7273 COMLST, COMLIST
7304 7304 MASK, . /ARRANGED IN COMMAND ORDER SAVING RELEVANT
/BITS FOR "TEST"
7305 0000 0 /NO FEWIND TYPE ERRORS
7306 0356 356 /READ
7307 7777 7777 /ILLEGAL COMMAND
7310 0246 246 /WRITE
7311 0246 246 /WPITE EOF
7312 0142 142 /SPACE
7313 1100 1100 /BACKSPACE
7314 6211 RANGET, CDF 10
7315 1736 TAD I XSTATUS /DATA TRANSFER IN PROGRESS?
7316 7640 SZA CLA
7317 5333 JMP REXIT /NO, SO SKIP THE TESTS
7320 6201 CDF
7321 1741 TAD I LORD /TEST FNEW FOR ADDRESSING OUT
/OF MAGTAPE ADDRESSING AREA

```

```

7322 7041      CIA
7323 1337      TAD FIRST
7324 7740      SMA SZA CLA
7325 5333      JMP REXIT      /<FIRST
7326 1741      TAD I LORD
7327 6211      CDF 10
7330 1340      TAD LAST
7331 7750      SPA SNA CLA
7332 4362      JMS XTEST      /< OR = LAST, SO WAIT UNTIL DONE
7333 6203      REXIT, CDF CIF
7334 5735      JMP I .+1
7335 7354      XFNEW+6
7336 7440      XSTATU, STATUS
7337 1777      FIRST, 1777      /EVENTUALLY 2325
7340 6001      LAST, -1777
7341 0046      LORD, 46
/
/MULTIPLIES INPUT ACC BY 3 AND EXITS WITH IT IN ACC
/
7342 0000      TIMES3, 0
7343 3361      DCA TEMP
7344 1361      TAD TEMP
7345 7104      RAL CLL
7346 1361      TAD TEMP
7347 5742      JMP I TIMES3
/
COMGO1=.
7350 7447      REWIND
7351 7500      GET
7352 7400      INITIALIZE
7353 7500      WRITE
7354 7453      WFOF
7355 7441      SPACE
7356 7441      BACKSPACE
7357 7473      CLEAR
/
CNTR,
7360 0000      COML, 0
7361 0000      TEMP, 0
/
/TEST ROUTINE FOR MAG TAPE STATUS
/
7362 0000      XTEST, 0
7363 7300      CLA CLL
7364 1736      TAD I XSTATUS      /IN PROGRESS
7365 7650      SNA CLA
7366 5364      JMP .-2
7367 6711      MTCR
7370 5367      JMP .-1
7371 6721      MTRP      /READY?
7372 5371      JMP .-1
7373 5762      JMP I XTEST
PAGE
/
/
/INITIALIZATION FOR MT/FOCAL
/
7400 4624      INITIA, TEST
7401 6203      CDF CIF
7402 4745      JMS I ARG      /PICKUP COMMAND WORD
7403 3236      DCA WORD
7404 1236      TAD WORD
7405 0216      AND K4      /PICKOUT BIT 9
7406 7650      SNA CLA
7407 1347      TAD BOMB
7410 3237      DCA MTSW

```

```

7411 1236          TAD WORD
7412 0220          AND K7303          /SAVE ONLY LEGAL BITS
7413 1217          TAD K404
7414 3236          DCA WORD
7415 5621          OK
7416 0004 K4,      4
7417 0404 K404,   404
7420 7303 K7303,  7303
/
/NORMAL EXIT TO FOCAL IS "OK"
/ERROR EXIT IS "SYNTAX" AND IS NOT RECOVERABLE
/
OK=JMP I .
7421 7263          EXIT
SYNTAX=JMS I .
7422 7252          ERP
X3=JMS I .
7423 7342          TIMES3
TEST=JMS I .
7424 7362          XTEST
/
/CONSTANTS AND VARIABLES
/
7425 7304 XMASK,  MASK
7426 3000 REWSTA, 3000          /TAPE REWINDING OF BOT
7427 7752 MTWC,   7752
7430 7753 MTCA,   7753
7431 7337 XFIPST, FIPST
7432 7340 XLAST,  LAST
7433 0001 COMM,   1
7434 6000 MAX,    -2000        /EVENTUALLY -2000
7435 2000 KMAX,   2000
7436 0404 WORD,   404          /TAPE 0, INTERRUPTS ENABLED, 200 BPI, ODD
7437 5576 MTSW,   MTFE        /DISABLE TAPE ERROR EXIT WHEN ZERO
STATUS,
7440 0001 MTIPSW, 1          /MAG-TAPE-IN-PROGRESS-SWITCH
/TPUE IF ZERO
/
/TAPE HANDLER COMMANDS
/
SPACE,
7441 4334 BACKSP,  JMS RWARG5
7442 0000          0
7443 4624          TEST
7444 1242          TAD .-2
7445 6201          CDF
7446 5330          JMP B01
7447 1240 REWIND,  TAD STATUS
7450 0226          AND REWSTATUS
7451 7640          SZA CLA          /EXIT IF BOT OR REWINDING
7452 5621          OK
MTEXEC,
7453 4624 WEOF,   TEST
7454 1233          TAD COMM
7455 1225          TAD XMASK
7456 3346          DCA COMB
7457 1746          TAD I COMB
7460 3346          DCA COMB
7461 1233          TAD COMM
7462 7104          RAL CLL
7463 7006          RTL          /COMMAND IN BITS 6-8
7464 1236          TAD WORD
7465 6714 B02,    6714        /LOAD COMMAND REGISTER, CLEARED IN MTINT
7466 7200          CLA
7467 3240          DCA MTIPSW
7470 1265          TAD B02

```

```

7471 6722          MTGO    /BITS 6-8 ARE FIELD 1 FOR BREAK EXTENSION
7472 5621          OK
7473 6732 CLEAR, 6732    /POWER CLEAR TAPE CONTROLLER
7474 7040          CMA
7475 3346          DCA COMB
7476 7001          IAC
7477 5262          JMP B02-3

GET,
7500 4334 WRITE,  JMS RWARGS
7501 0000 COUNT,  0
7502 4334      JMS RWARGS
7503 0000 IA,     0
7504 4624          TEST
7505 1303          TAD IA
7506 1234          TAD MAX
7507 7500          SMA
7510 4622          SYNTAX /INITIAL ADDRESS TOO BIG ?30.72
7511 1301          TAD COUNT
7512 7500          SMA
7513 4622          SYNTAX /FINAL ADDRESS TOO BIG ?30.75
7514 1235          TAD KMAX
7515 7041          CIA
7516 3632          DCA I XLAST
7517 1303          TAD IA
7520 3631          DCA I XFIRST
7521 6201          CDF
7522 1303          TAD IA
7523 4623          X3
7524 1377          TAD M1
7525 3630          DCA I MTCA
7526 1301          TAD COUNT
7527 4623          X3
7530 7041 BO1,    CIA
7531 3627          DCA I MTWC
7532 6211          CDF 10
7533 5253          JMP MTEXEC

/
/
/PEAD-WRITE ARGUMENT SUBROUTINE
/   JMS RWARGS
/   STORAGE FOR (POSITIVE) RESULT
/   RETURN
/

7534 0000 RWARGS, 0
7535 6203          CIF CDF
7536 4745          JMS I ARG
7537 7000          NOP
7540 7510          SPA
7541 4622          SYNTAX /POSITIVE ARGUMENTS ONLY! ?30.97
7542 3734          DCA I RWARGS
7543 2334          ISZ PWARGS
7544 5734          JMP I RWARGS

/
/

7545 5755 AFG,    ARGUMENT
7546 0000 COMB,   0

/
/
/INTEFRUPT HANDLER FOR TC-58
/
/

*7547
7547 5576 BOMB,   MTFE
7550 0000 MTINT,  0
7551 6703          MTSF
7552 5372          JMF MTEXTIT    /NOT TAPE

```

```

7553 6706      MTIS
7554 3240      DCA STATUS
7555 6706      MTES
7556 0346      ANL COME      /MASK STATUS
7557 3375      DCA COMEED    /SAVE FOR FADC
7560 1375      TAD COMEED
7561 7650      SNA CLA
7562 5371      JMP MTEXIT-1   /OK
7563 1237      TAD MTSW      /EPROF, TEST EXIT SWITCH
7564 7450      SNA
7565 5371      JMP MTEXIT-1   /INHIBIT EXIT
7566 6201      CLF
7567 3776      DCA I ZEF0    /EPROF DIAGNOSTIC SOON
7570 2350      ISZ MTINT     /SKIP OVER FMF@ RETURN+1
7571 6712      MTAF      /MASH DOWN THE FLAG
7572 6203      MTEXIT, CDF CIF
7573 5750      JMP I MTINT    /AND RETURN
7574 0012      12      /EXP FOR FADC
7575 0000      COMEED, 0      /HOLD FLAC FOR FADC
7576 0000      ZEF0, 0      /LOAD FLAC FOR FADC
7577 7777      M1,      -1
              /$
              /
              /$

```

ARG	7545
ARGUME	5755
BACKSP	7441
BOMB	7547
B01	7530
B02	7465
CHAR	0142
CLEANU	5772
CLEAR	7473
CNTR	7360
COMB	7546
COMBED	7575
COMGO	1171
COMGO1	7350
COML	7360
COMLIS	7273
COMLST	7303
COMM	7433
COUNT	7501
DIAG	7262
DIFF	7272
EFUN3I	0100
ERR	7252
ERROR4	4526
EVAL	1603
EXIT	7263
FIPST	7337
FLAC	0044
FNEWTE	0000
FNTABF	0376
GET	7500
GETC	4506
GLIST	1406
HINBUF	0037
IA	7503
INITIA	7400
INTEGE	0052
KMAX	7435
K4	7416
K404	7417
K7303	7420
LAST	7340
LIMIT	2334
LORD	7341
MASK	7304
MAX	7434
MCOM	7375
MCOMMA	5771
MOST	7373
MTAF	6712
MTCA	7430
MTCR	6711
MTEXEC	7453
MTEXT	7572
MTFE	5576
MTGO	6722
MTINT	7550
MTIPSW	7440
MTLC	6716
MTRC	6724
MTPS	6706

MISF	6703
MTSW	7437
MTTP	6721
MTWC	7427
M1	7577
OK	5621
OKEXIT	7267
PNTF	0163
POP'A	1413
POPJ	5502
PTF	0163
PUSHA	4503
PUSHJ	4501
FANGET	7314
FFVIND	7447
FEVSTA	7426
FEYIT	7333
FVAFGS	7534
SOFTC	4511
SOFT1J	7225
SPACE	7441
SPNOP	4521
STATUS	7440
SYNTAX	4622
TAPF	5571
TFMP	7361
TFPM	1410
TFST	4624
TIMES3	7342
T1	5575
T2	6177
WFOF	7450
WORD	7436
W117F	7500
XADC	7571
XEOMB	0165
XCLEAN	7261
XCOMM	7271
XEFFOF	7270
XFGET	7376
XFIFST	7431
XFNEW	7346
XFNEW2	7154
XFNEW3	7554
XLAST	7432
XMASK	7425
XMTINT	0162
XPFOC	0164
XSTATU	7336
XTEST	7362
XT1	6171
XT2	6571
X3	4623
ZEFO	7576