



DECUS

PROGRAM LIBRARY

DECUS NO.	8-298
TITLE	OCTMON - AN OCTAL MONITOR FOR THE PDP-8 COMPUTER
AUTHOR	Peter Lemkin
COMPANY	National Institute of Health Bethesda, Maryland
DATE	June 3, 1969
SOURCE LANGUAGE	MACRO

DECEMBER 1954



OCTMON - AN OCTAL MONITOR FOR THE PDP-8 COMPUTER

1.0 ABSTRACT:

OCTMON is an octal debugging monitor for the PDP-8 that facilitates debugging for machines with several memory fields. It is relocatable within a field, and may be put in any field. It uses 1400 (octal) locations and 13 (octal) locations in page 0 of its resident field. It also uses 11 (octal) locations of any other field in which there is a breakpoint. Only one restorable breakpoint is allowed at any time and may be in any field. There are options to dump out sections of memory, punch and read bin tapes, enter octal numbers into memory, start the program being tested, and enter a number into the MQ. The monitor will work without EAE, extended memory and without DECTAPE/DISC systems.

1.1 REQUIREMENTS:

OCTMON requires 1400 (octal) memory locations and 13 (octal) page 0 locations in the resident field, and 11 (octal) page 0 locations in any other field with breakpoints. For the high version locations, <6200, 7577> and <165, 177> are in the resident field 0, and <165, 175> in the other fields are used. The starting address is 6200. For the low version locations, <2000, 3577> and <165, 177> are in the resident fields and <165, 175> in the other fields are used.

2.0 DESCRIPTION:

The octal monitor is a generalized debugging package and may be used in the usual way by assigning a breakpoint and then starting the program. However, this debugging monitor allows easier communication between fields by giving the operator easy access to all the fields. This is done by use of the "F"(set field register) command which sets a field register to the value of the low order digit in the buffer. This then lets the user investigate the current field as assigned by "F". He may print out memory, insert into memory, punch and read binary tapes, insert breakpoints, and start a program in the current field. The command structure is such that the user can debug faster and more easily than with DDT.

A command is a single keyboard character. Appendix 1 lists the 22 commands. When a command is typed, the monitor obeys it immediately and does not wait for a carriage return. If an illegal command is typed, a "?" is typed followed by a "carriage return, line feed, *". The * signifies that the monitor is waiting for a command.

The commands "space," "P" and octal characters do not cause the "carriage return, line feed, *".

2.1 SOFTWARE REGISTERS:

There are three software registers which are used quite frequently in the monitor operation. These are the B or buffer register, the P or pointer register, and the FLD or memory field register.

B Register: The B, buffer, register is the four octal digit input register. It accepts octal characters from the keyboard and shifts the current octal number left one octal digit (losing the old high order octal digit) to make room for the new low order octal digit. If an error is made, just type the four digit octal number in again.

Example: = 1234
Now, type in a 5
Then the new contents of B is
= 2345

P Register: The P, or pointer, register is a four octal digit memory address pointer register. When a command needs to reference memory it looks at this register for a pointer.

FLD Register: The FLD, or memory field, register is a four octal digit field pointer register. The 8'S digit gives the current operating field that the memory address register P is pointing to.

The command set references these registers in the different operations.

Example: To dump 1235 to 1734 of field 1, type:
*1235Z -set up lower bound
*1F -set up memory field pointer
*1734L -set up upper bound, dump in line format.

To examine any of these software registers:

Command "V" prints out the contents of B.
Command "W" prints out the contents of P.
Command "E" prints out the contents of FLD.

2.2 BREAKPOINT:

The user may stop his program at a selected point (called a breakpoint) by use of the command "B". This command will set up a breakpoint in the current field (as determined by the FLD register) and at the memory location pointed to by P.

When a breakpoint occurs in, for example, field N (with the monitor in a field M), the breakpoint causes a jump to a dispatching subroutine in page 0 of the field N. This subroutine saves the AC, and puts the breakpoint field in the AC. It then returns to the monitor which reads the break field, gets the AC, and prints out:

L ##### A ##### M ##### P ##### F

where:

L is the link bit at the time of break
A is the ACC at time of the break
M is the MQ (if exists) at the time of the break
P is the address of the breakpoint at the time of the breakpoint
F is the field in which the breakpoint occurred. (8'S digit).

Note: # means octal digit

Therefore, if multiple breakpoints are left in various fields, the flow of control between fields can be easily checked.

2.3 OCTAL READ IN:

Another useful feature of the octal monitor is the method of reading octal numbers into memory. After an octal number is typed, a carriage return puts it into memory at the location given in the pointer P. It then increments the pointer by 1. Since the command line feed is ignored, an octal program can be loaded this way if a field register load and address register pointer load are included in the beginning of the tape. "SHIFT/M" command can be used to punch out this type of formatted paper tape. It generates the following format:

```
#F####Z####CARRIAGE RETURN LINEFEED  
#####CARRIAGE RETURN LINE FEED  
.  
.  
.
```

Note: Any character which is not a command or number is ignored and returns control to the monitor loop.

2.4 OCTAL DUMPS:

There are three different formats to print octal dumps using the monitor. All three use the same set up procedure. Put the starting address of the dump in the P register and the final address in the buffer B.

Command "M" will then print out the value of P and the contents of the memory location P points to. It then skips a line, increments P by 1, and prints out the same thing for P+1. This continues until P>B when control is returned to the monitor loop.

Command "SHIFT/M" which was discussed previously punches an ASCII octal paper tape. This enables the loading of ASCII octal paper tapes into various fields.

Command "L" prints the value of the pointer and then the contents of the 10 (decimal) consecutive locations starting at P. It then skips to the next line and repeats, except printing P+10 (decimal) until P>B.

Note: To stop printing in any of these modes hit any character to return to the monitor command loop.

2.5 RESTRICTIONS:

A restriction on the breakpoint is that it must contain an executable instruction. The assigning of multiple pseudo breakpoints is possible except that only the last one assigned will be restored when the command "U" is executed (i.e., there is no pushdown list).

If the octal monitor is used with a disc or tape system that is resident in the last page of field 0, <7600, 7777>, the command "T" will be functional (exit to index), otherwise it might bomb the monitor.

A change can be made in the monitor to eliminate this problem. Then, typing a "T" will cause a "?" to be typed as if the "T" were undefined and the system will not bomb.

Changes: High version, i.e. <6200, 7577>.
Change contents of 6256 from 6605 to 7441.
(If changing the symbolic MACRO-8 tape, change "T" in the Subroutine Pointer Table to "QUES-1").

Low version, i.e. <2000, 3577>.
Change contents of 2056 from 2405 to 3241.

The command "Q" (move the buffer into the MQ register) will be a NOP if there is no extended arithmetic.

There is no octal monitor protection on the commands "carriage return," "space," "P," and "X" which could enter data into the monitor core area.

The instructions on page 0 will be generated in using the monitor so that page 0 need not be saved if the octal monitor is to be loaded by a tape or disc system.

2.6 STARTING UP:

Load the octal monitor into core. Then "LOAD ADDRESS" the starting address (6200 for high ((standard)) version, or 2000 for low version) and hit "START." The monitor should give you a "*" then carriage return line feed. It is now ready to accept commands. The first command you should do to set up the octal monitor is the assign field or the "F" command to the desired field. This should be done before anything else. The monitor is now ready for further commands.

3.0 ADDITIONAL COMMANDS (R+D):

There are currently two unimplemented instructions. The user could write his own subroutines and link these with the octal monitor by the following format:

For the low version $\langle 2000, 3477 \rangle$ subtract 4200 from all addresses.

Monitor location 6266: User subroutine address 1 /command R
Monitor location 6267: User subroutine address 2 /command D

The user can return to the monitor by a JMP I SUBROUTINE or a JMP 6200. For both the high and low versions the:

P register is location 176
B register is location 175
FLD register is location 174.

```

/          APPENDIX 1.
/"OCTMON" AN OCTAL MONITOR FOR THE PDP-8.
/VERSION 10.R
/PETER LEMKIN, JUNE 3, 1969
/NATIONAL INSTITUTES OF HEALTH
/THE IDEA FOR THIS PROGRAM WAS TAKEN FROM
/NATL. BUREAU OF STANDARDS TECH , MEMO 650,01/1 2/4/69
/**A HEXADECIMAL MONITOR SYSTEM FOR THE INTERDATA COMPUTER**
/ BY P. STEIN.
/ THIS MONITOR FACILITATES THE MANIPULATION OF
/ BINARY MANUSCRIPTS FROM THE TTY.
/ STORAGE REQUIREMENT: <165,177>, AND <6200,7577> FIELD 0.
/   ALSO <165,175> OF THE OTHER FIELDS THAT HAVE
/   BREAK POINTS IN THEM,
/   THE PAGE 0 AND MONITOR STORAGE ARE RELOCATABLE WITHIN
/   A FIELD BY CHANGING THE ORIGIN=6200, AND THE PAGE 0
/   ORIGIN *165.
/ IT USES THE EAE, EXTENDED ARITHMETIC, "LOOKING AT THE MQ".
/ NOTE: NEITHER EAE OR EXTENDED MEMORY IS "NECESSARY",
/ THE STARTING ADDRESS IS : 6200, IN FIELD 0.(FOR THE VERSION SUPPLIED),
/ THE MONITOR IS RELOCATABLE IN OTHER FIELDS BY
/ MAKING CHANGES ON PAGE 2 OF THE SYMBOLIC PROGRAM,
/   CHANGE FIELD 0 TO FIELD "N". CHANGE "CDF=6201"
/   TO "CDF=62N1", AND CHANGE "CIF=6202" TO
/   "CIF=62N2",
/   WHERE: "N" IS THE NEW FIELD.
/ THERE ARE THREE SOFTWARE REGISTERS:
/   P, THE MEMORY ADDRESS POINTER
/   B, THE INPUT BUFFER REGISTER
/   FLD, THE MEMORY FIELD POINTER REGISTER
/ ALL OCTAL INPUT WHEN TYPED IN GOES INTO THE B REGISTER FROM
/ WHICH IT CAN BE MOVED INTO THE P (BY "Z" COMMAND) OR FLD (BY
/ "F" COMMAND) REGISTERS.
/
/ COMMANDS*****
/ >: GIVES CR,LF , WAITS FOR NEXT COMMAND
/ Z: LOADS B INTO P, ZERO'S B, <CR,LF>.
/ CR: "CARRIAGE RETURN", B INTO MEM<P>,P+1 INTO P, ZERO'S B, <CR,LF>.
/ SP: "SPACE", B INTO P, ZERO'S B, THERE IS NO <CR, LF>.
/ P: MOVE B INTO MEM<P>,P+1 INTO P,NO <CR,LF>, DO NOT CLEAR B.
/ Q: MOVE THE BUFFER B INTO THE MQ REGISTER, ZERO B,<CR,LF>.
/ LF: "LINE FEED", NO OPERATION EXCEPT LINE FEED.
/ I: INCREMENT P, P+1 INTO P, ZERO'S B,<CR,LF>.
/ R0: "RUBOUT", DECREMENT P, P-1 INTO P,ZERO'S B, <CR,LF>.
/ W: PRINT POINTER P, <CR,LF>,
/ V: PRINT BUFFER B, <CR,LF>
/ CONTROL/G: JUMP TO THE LOCATION GIVEN IN THE BUFFER
/   AND IN THE FIELD GIVEN BY FLD, (THE FIELD FLAG).
/ M: PRINTS CONTENTS OF MEMORY FROM LOC, P TO B,
/   TO STOP TYPING AT ANY TIME, TYPE ANY CHARACTER,
/   TO SUPPRESS PRINTING OF THE POINTER EACH TIME
/   TYPE SHIFT/M INSTEAD OF M.
/   THIS WILL ALSO GENERATE LEADER TRAILER CODES
/   THE VALUE OF THE POINTER, FOLLOWED BY THE LETTER Z
/   TO ALLOW READING THE DUMP BACK IN.
/ L: PRINT THE CONTENTS OF MEMORY LOC , P
/   TO LOC, B IN STEPS OF 10 WORDS/LINE,
/   TO STOP TYPING AT ANY TIME, TYPE ANY CHARACTER,
```


/ T: TRANSFER CONTROL TO DECTAPE/DISK SYSTEM @7600, FIELD 0. .
/ (THIS CAN BE DONE FROM ANY FIELD.)
/ THE PUNCH WILL BE DONE IN THE CURRENT FIELD , "FLD" .
/ K: PUNCH BINARY TAPE FROM MEMORY LOCATIONS P TO B WITH LDR./TRLR,
/ X: READ BIN TAPE INTO MEMORY, RETURN TO MONITOR WHEN DONE.
/ SET RIGHT SWITCHES = NEGATIVE NUMBER FOR LOW SPEED PAPER
/ TAPE READER, A POSITIVE NUMBER FOR HIGH SPEED PAPER TAPE READER,
/ F: SET THE FIELD FLAG "FLD" = CONTENTS OF BUFFER, (1'S DIGIT),
/ E: PRINT OUT THE CONTENTS OF THE FIELD FLAG, "FLD", (8'S DIGIT),
/ B: INSERT A BREAK INTO MEM<P>, SAVE P, SAVE OLD MEM<P>.
/ THEN DO <CR,LF>, RETURN TO MONITOR LOOP.
/ EXECUTION OF THE BREAK INSERT WILL CAUSE CONTROL
/ TO BE TRANSFERED BACK TO THE MONITOR AND THE
/ CONTENTS OF THE LINK BIT, AC, MQ, P OF THE BREAK,
/ AND FIELD OF THE BREAK TO BE PRINTED OUT.
/ U: UNBREAK. RESET THE INSTRUCTION THAT WAS SAVED
/ WHEN LAST DID BREAK, ALSO LAST P SAVED (WITH B COMMAND)
/ IS ENTERED INTO P.
/ NOTE: OTHER COMMANDS R AND D WILL BE ADDED LATER, PRESENTLY
/ THEY ARE NOP'S.
/

