```
               DDDDD   EEEEE   BBBBB   U    U  GGGGG
               D    D  E       B    B  U    U  G
               D    D  E       B    B  U    U  G
               D    D  EEE     BBBBB   U    U  G
               D    D  E       B    B  U    U  G   GG
               D    D  E       B    B  U    U  G    G
               DDDDD   EEEEEE  BBBBB   UUUUUU  GGGGGG


               888888  000000  888888  000000
               8    8  0    0  8    8  0    0
               8    8  0    0  8    8  0    0
               888888  0    0  888888  0    0
               8    8  0    0  8    8  0    0
               8    8  0    0  8    8  0    0
               888888  000000  888888  000000  V1.3
```

WRITTEN BY:
IAN KETTLEBOROUGH

COPYRIGHT (C) 1977 BY PROCESSOR TECHNOLOGY CORP.

PURPOSE: This program is used as an aid in debugging programs which are under development. It provides the ability to set multiple breakpoints. When a breakpint is encountered, the registers, CPU flags and memory locations can be examined and modified. Program execution can then be continued from the breakpoint.

There are two versions of DEBUG on the PTDOS system disk.

DEBUG3 - loads and runs at 3000H

DEBUG5 - loads and runs at 5000H

There is no difference between the two programs except for the memory space used. This allows the debugging of a program that uses the same memory space as one or the other of the two debug programs. The debug program uses a little over 4K of memory, and may be restarted at its starting address.

SUMMARY OF COMMANDS:

BXXXX	SET break point at address XXXX
DXXXX	DELETE breakpoint at addr XXXX
T	Type out ALL currently set breakpoints
K	Delete ALL currently set breakpoints
H	Set output mode to HEX
I	Set output mode to INSTRUCTION
C	Set output mode to CHARACTER
XY	Display CPU register Y
RXXXX	Begin execution at address XXXX
A	Set STATIC breakpoint mode
Z	Set REMOVE breakpoint mode
PXXXX	PROCEED from a breakpoint
V	COMPLEMENT output to VDM mode
L(FILE)	Complement output to PTDOS file
E	EXIT to monitor
WXXXX,YYYY	DUMP memory from address XXXX to address YYYYY
FXXXX,YYYY,XX	FILL memory from address XXXX to YYYY with XX
SXXXX,YYYY,BB,MM	SEARCH MEMORY starting at address XXXX continuing to address YYYY for byte BB using mask MM
EXPR/	EXAMINE memory location specified by expression
EXPR=	OUTPUT value of EXPR in HEX
EXPR#	OUTPUT value of EXPR in DECIMAL

COMMAND DESCRIPTIONS

All commands EXCEPT those that start with an EXPRESSION must be preceeded by an ESCAPE (ESC or ALT MODE key). This is echoed as a dollar sign ($). ALL commands must be terminated with a carriage return EXCEPT when the command ends with a # or =.

SET BREAK POINT (BXXXX)

A BREAKPOINT is set at address XXXX. IF the breakpoint is part of a multi-byte instruction it must be set on the first byte of that instruction.

DELETE BREAKPOINT (DXXXX)

This command DELETES the breakpoint currently set at address XXXX.

CHANGE OUTPUT MODE TO HEX (H)

This command set the output mode for the examine memory to HEX.

CHANGE OUTPUT MODE TO INSTRUCTION FORMAT (I)

This command sets the output format for either the W command or EXAMINE memory to to INSTRUCTION format.

CHANGE OUTPUT MODE TO CHARACTER (C)

This command sets the output mode for the examine memory to the CHARACTER mode. If the memory location this is to be printed is a printable ASCII character, then that character will be printed. Otherwise it will be printed in HEX.

EXAMINE CPU REGISTERS AND FLAGS (XY)

This command is used to EXAMINE AND MODIFY the CPU registers and FLAGS after a breakpoint is encountered. Y in the command can be one of the following letters to designate the register:

A -- Accumulator
B -- Register B
C -- Register C
D -- Register D
E -- Register E
F -- CPU flags
H -- Register H
L -- Register L
M -- Byte POINTED to by HL
P -- Proceed address from a breakpoint
S -- Stack pointer on entry to breakpoint

CARRIAGE RETURN for Y will print out all of the registers.

After a register has been examined, it can be modified, the next register can be examined or the command mode can be re-entered.

If a NEW VALUE is typed after the register is displayed, the new value will replace that currently in the register. The delimiter on the new value must be either a SPACE to indicate that the next register should be displayed or a CARRIAGE RETURN to reenter the command mode.

If a new value is not to replace that of the register, then either a SPACE or CARRIAGE RETURN must be entered. This will have the same effect as above.

All of the registers are printed in NUMERIC format except for the CPU flags which are printed as letters to identify those flags which were set when the breakpoint occured.

The letters are:

S -- Sign flag
L -- Zero flag
A -- Auxiliary carry flag
P -- Parity flag
C -- Carry flag
N -- No flags set

If NO flags are set, then 'F=N' will be printed. Also, to replace the flag value, one or more of the above letters can be entered to form the desired flag setting.

RUN COMMAND (RXXXX)

This command will START program execution at address XXXX. The CPU registers will be in an unknown state at that time.

SET STATIC BREAKPOINT MODE (A)

This command seet the BREAKPOINT MODE such that breakpoints are NOT deleted when they are encountered. This mode is present when DEBUG is first run.

SET REMOVE BREAKPOINT MODE (Z)

This command sets the BREAKPOINT MODE such that breakpoints ARE deleted when they are encountered.

PROCEED FROM A BREAK POINT (PXXXX)

Entering this command will cause EXECUTION of the program to continue with the instruction that caused the break. All CPU registers will be reloaded as they were when the beakpoint was encountered (or as modified using the X command). The XXXX is optional and is used to specify how many times the breakpoint is to be encountered before returning control to the DEBUGGER. If not specified, the default is one (1) time.

COMPLEMENT VDM OUTPUT (V)

This command COMPLEMENTS the VDM output option. That is, if output were going to the VDM, after a V command it would be to the PTDOS CONOUT routine.

COMPLEMENT OUTPUT TO LPT (LFILE)

This command will complement the output to FILE flag. If set, then all output will be to the VDM and PTDOS file LFILE. The file is not opened util the first time the L command is used. If an error occurs when the file is opened, the PTDOS error code will be printed (in hex) followed by a question mark. The output file MUST exist or an error will be printed.

EXIT TO PTDOS (E)

This command terminates execution of the DEBUGGER and returns to PTDOS.

SEARCH MEMORY (SXXXX,YYYY,BB,MM)

This command SEARCHes memory starting at address XXXX and continuing to address YYYY. As each byte is examined it is first ANDed with the MASK MM and then checked for equality to the SEARCH BYTE BB. If it is equal, then the memory address and the byte at the address are printed. The mask will default to 0FFH if not specified.

WRITE MEMORY (WXXXX,YYYY)

This command causes a section of memory to be WRITTEN out in the CURRENT OUTPUT MODE. If the output mode is NOT instruction format, then memory will be dumped with 14 bytes displayed on each line. The bytes are first output in hex, followed on the right by a CHARACTER dump of the same 14 bytes. If the byte represents a printable character in the range of BLANK to SQUIGGLE (', HEX 7E) the byte will be output as the printed character. If it is not in this range a PERIOD will be printed in its place.

In the INSTRUCTION mode, memory will be dumped one instruction per line in decoded format.

Memory will be dumped starting at XXXX and continuing to YYYY. Address YYYY need NOT be specified and if absent will be taken
/////////
The memory dump can be terminated at the end of a line by pressing MODE or control/@.

FILL MEMORY COMMAND (FXXXX,YYYY,XX)

This command will FILL memory between XXXX and YYYY with the byte XX.

EXAMINE MEMORY (/)

If an EXPRESSION is entered that is followed by a SLASH (/) then the memory location specified by that expression will be displayed in the current output mode.

Several options are not available after the location is displayed.

CARRIAGE RETURN	to return to command mode
SINGLE QUOTE ()	to output the location in CHARACTER mode.
DOUBLE QUOTE (")	followed by a character to specify a character replacement for that location.
SEMI-COLON (;)	to output the byte in INSTRUCTION format.
UP ARROW (")	to examine the previous location
LINE FEED	to examine the next location.
EQUAL SIGN (_)	to output the location in HEX.
COLON (:)	to indicate INSTRUCTION input.

If a REPLACEMENT is specified, it must be followed by a CARRIAGE RETURN, LINE FEED, or UP ARROW. These function as specified above.

XXXX, YYYY OR EXPR

XXXX (or YYYY) can be ANY EXPRESSION as defined below. The expressin can involve the operatiors +,-,*, and % (DIVIDE). A period (.) in the expression represents the address of the LAST memory location that was examined. If a number is preceeded by an EXCLAMATION POINT (t), then is is interpreted as a decimal number. The expression can be terminated by one of the three following characters:

* -- output value of expression in HEX
# -- output value of expression as an
     unsigned 16 bit DECIMAL number
/ -- EXAMINE the specified memory location

The replacement value of the EXAMINE REGISTER or EXAMINE MEMORY command can also be an expression but, it must be terminated by a CR, LF or UP ARROW. Note that the PERCENT SIGN (%) is used for DIVIDE as the SLASH is used in the EXAMINE command. Multiply and divide operate on two 16 bit unsigned numbers. Overflow or divide by zero are not checked for. With division, the result is the quotient, the remainder is not available.

INSTRUCTION FORMAT OUTPUT

If the INSTRUCTION FORMAT output is specified (using either the I command or ";") the memory location will be decoded into the appropriate 8080 Instruction mnemonic. The 12 unimplemented instructions of the set are output in hex. If the instruction is MULTI-BYTE (i.e. JMP), then the appropriate number of bytes to complete that instruction will be output in HEX instead of decoding them into instruction format. This does not apply if the up arrow is used in the EXAMINE command.

INSTRUCTION INPUT

After an EXAMINE (/) command, a COLON (:) can be entered to indicate that what follows is an instruction. The standard INTEL mnemonics have been implemented for direct input. "P" is used instead of 'PSW' and "S" instead of 'SP'.

The instruction must immediately follow the colon and, if no operands are required, teminate with a CR, LF, UP ARROW or BLANK. Operands, when present, are separated form the op-code by one blank only. If two operands are required, the must be separated by a COMMA. The instruction will be stored in memory at the currently examined location, taking as many sucessive locations as necessary for the instruction.

DEBUG INITIALIZATION

When DEBUG 8080 is first started, it will request which of the eight restart locations is to be used for the breakpoint. Any number from 0 to 7 is a valid response. When a breakpoint is set at an address, the byte at that location is replaced by a RST x, where "x" was specified as above.

Any time that the DEBUG is available for a command to be entered, a ">" prompt will be output. When a breakpoint is encountered during program execution an asterisk (*) will be printed, followed by the breakpoint address. If output is to the VDM then all of the registers will be printed out when the breakpoint is encountered. The prompt will then be output and DEBUG commands can be entered. If necessary the debugger can be manually restarted at address 3000H or 5000H depending on whether DEBUG3 or DEBUG5 was loaded.

The DEBUGGER contains its own VDM-1 output driver. When the DEBUGGER is first started output is set to go to the VDM. This can be changed by using the V command to direct output to the system CONOUT driver instead.

The VDM driver incorporates a variable SPEED options which is set by striking a key (0-9) while output is being displayed. Output can be temporarily suspended by pressing the space key. Output continues when any other key beside the SPACE key is hit. The speed is initially set to 2.

The maximum number of breakpoints that can be set at one time is 12.