

QUICK & DIRTY DEBUG FOR THE PC

I. Notational Conventions

address = segment:offset; debug recognizes segment register names (i.e. cs, ds, ss, es.)
value = 1 - 4 digit hexadecimal
range = address,address OR address L value Note L means "length"
list = series of byte values
[] denotes parameter is optional

II. Commands

Syntax	Meaning	Example
1. a [address]	Assemble code. Debug prompts for each instruction. Enter empty line to terminate.	-a 11F1:0100 mov ax,2 11F1:0103 add ax,3 11F1:0106
2. c range address	Compare portion of memory Results displayed as : addr1 byte1 byte2 addr2	-c 100 L10 200
3. d [range]	Dump contents of memory	-d ds:0100 L 10
4. e address [list]	Enter bytes at address Use space and hyphen to advance and backup. Hit (Enter) to quit e command	-e cs:0100 11FF:0100 F1.
5. f range list	Fill bytes starting at address using list	-f ds:0100 L10 00
6. g [=address] [addresses]	Execute program starting at first address; breakpoints given in second list	-g CS:0000 09
7. h value value	Perform hexadecimal arithmetic Displays sum and difference	-h 02A1 009E
8. i value	Input byte from port	-i 2F8
9. l [address]	Load file whose name is given by n command into memory starting at address. Note address ignored with .EXE files	-l
10. m range address	Move block of memory	-m 100 L10 200
11. n filename	Sets Name of file for use by l or w commands	-n prog0401.exe
12. o value byte	Output byte to port	-o 2F8 42
13. p [=address] [value]	Proceed is like trace but will return from INT instructions	-p CS:0100 10
14. q	Quit	-q
15. r [reg name]	Display/Change Register contents With register, Debug returns current value and prompt for new	-r IP IP 0100 :
16. s range list	Search range of bytes for values in list	-s cs:0100 L10 00
17. t [=address] [value]	Execute the next n instructions (n is value) starting at address DO not trace INT instructions.	-t
18. u [range] [Address]	Unassemble code	-u l 10
19. w	Write to file. Number of bytes must be in BX:CX pair	-w