Short Report on Debug Tutorial

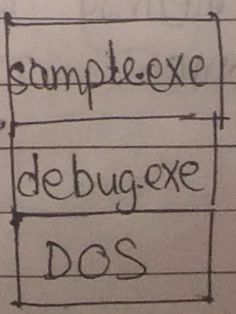
A debugger displays the contents of the memory and lets us view registers and variables as they change. It can be used to test assembler instructions, try out new programming ideas, or to carefully step through your programs.

Functions of a debugger

- Assemble short programs
- View programs source code along with its machine code.
- View the CPU registers and flags
- Trace or execute a program, watching variables for changes
- Enter new values into memory.
- Search for binary and ASCII values in memory
- Move a block of memory from one location to another
- Fill a block of memory
- Load and write disk files and sectors

Command To debug a sample program

- debug sample.exe



Debug Commands

Program Creation and Debugging

Assemble program using inst. mnemonics (A).

Execute program in memory (G).

Display contents of registers & flags (R)

Proceed past an instruction/loop (P)

Trace a single instruction (T)

Disassemble memory into mnemonics (U)

Memory Manipulation

Compare memory ranges (C)

Display (Dump) content of memory (D)

Enter bytes into memory (E)

Fill a memory range with single value (F)

Move bytes from one memory range to another (M)

Search a memory range for specific values (S)

Miscellaneous

Perform hexa decimal Add^n and Sub^n (H)

Quit Debug and return to DOS (Q)

Input-Output

Input a byte from port (I)

Send a byte to port (O).

Load: data from disk (L)

Write data from memory to disk (W)

Create a file name for use by the L and W commands (N).

Default Values

1. All segment registers are set to the bottom of free memory, just above debug.exe program.
2. IP is set to 0100h
3. 256 bytes of stack space is reserved at the end of current segment by debug.
4. All of available memory is allocated.
5. BX:CX are set to the length of the current program or file.
6. Flags
   - NV (overflow flag clear)
   - UP (Direction flag = UP)
   - EI (Interrupts enabled)
   - PL (sign flag = positive)
   - NZ (Zero flag clear)
   - NA (Auxillary Carry flag clear)
   - PO (odd parity)
   - NC (Carry flag clear)

Command Parameters

Address eg. F000:100, DS:200, 0AF5

Filespec eg. file1, C:\asm\progs\test.com

List eg. 10,20,30,40; 'A','B',50

Range eg. format 1: address,[address] 100,500

Sector String Value

'COMMAN' eg. 3A, 3A6F.

Important Commands

① assemble A [address]

Assemble a program into machine language

eg. A 100 Assemble at CS:100h
A Assemble at current location
A DS:2000 Assemble at DS:2000h

② compare C range address

Compares bytes between a specified range with the same number of bytes at a target address

eg. C 100 105 200 Bytes betn DS:100 and DS:010
are compared to bytes at DS:0200

③ dump D [range]

displays memory on the screen as single byt
in both hexadecimal and ASCII

eg. D 150 15A dump DS:0150 through 015A

④ enter E address [list]

Place individual bytes in memory on supplying starting memory location

eg E CS:100 "This is a string"

⑤ fill F range list

Fills a range of memory with a single value or list of values.

eg. F CS:300 CS:1000, FF Fill locations CS:300 through CS:1000 with hex FFh.

⑥ G10 G1 [=address] [addresses]

Execute the program in memory.

eg. G Execute from current location to end of program.

G1=10 50 Begin execution at CS:10 and stop before the instruction at offset CS:50.

⑦ hex H value1 value2

Performs addition and subtraction on two hexa-decimal numbers.

eg. H 1A 10 Display 2A 0A.

⑧ input I port

inputs a byte from a specified input/output port and displays it in hexadecimal.

eg. -I 3F8 display 00

⑨ load L [address] [drive] [firstsector] [number]

loads a file (or logical disk sectors) into memory at a given address

eg. L 100 2 A 5 load five sectors from drive C, starting at logical sector number 0A11.

⑩ move M range address

copies a block of data from one memory location to another

eg. M 100 105 1100 Move bytes in the range DS:100-105 to location DS:110

⑪ Name N [pathname] [arglist]

initialize filename in memory.

eg. N b:myfile.dta

⑫ Output O port byte

outputs a byte to a specified port

eg O 3F8 00

(13) proceed P [=address] [number]

executes one or more instructions / subroutines

eg P=150 6 execute 6 instructions starting at CS:0150

(14) Quit Q

quits debug and return to DOS

(15) R (Register) R [register]

display register and flag contents, allowing them to be changed.

eg. R display contents of all registers
R F display all flags and prompt for a new flag value.

(16) search S range list

searches a range of addresses for a sequence of one or more bytes.

eg. S 100 1000 0D search DS:0100 to DS:1000 for the value of 0Dh

(17) trace T [=address] value

eg T=105 10 → hex → Trace 16 instructions Starting at CS:105

(18) Unassemble U [range]

translate memory into assembly language mnemonics.

eg U 100 108 disassemble bytes from CS:100 to CS:108

(19) write w [address] [drive] [firstsector] [number]

write a block of memory to a file or individual disk sectors.

eg W 100 0 02 Write two sectors to drive A from location CS:0100 starting at logical sector number 0.