```
someone@debian-box:~/Documents/assembly/eatsyscall$ gdb ./eatsyscall
GNU gdb (Debian 7.7.1+dfsg-5) 7.7.1
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./eatsyscall...done.
(gdb) list
1     ;  Executable name : EATSYSCALL
2     ;  Version         : 1.0
3     ;  Created date    : 1/7/2009
4     ;  Last update     : 2/18/2009
5     ;  Author          : Jeff Duntemann
6     ;  Description     : A simple program in assembly for Linux, using NASM
2.05,
7     ;    demonstrating the use of Linux INT 80H syscalls to display text.
8     ;
9     ;  Build using these commands:
10    ;    nasm -f elf -g -F stabs eatsyscall.asm
(gdb)
11    ;    ld -o eatsyscall eatsyscall.o
12    ;
13
14    SECTION .data                ; Section containing initialised data
15
16          EatMsg: db "Eat at Joe's!",10
17          EatLen: equ $-EatMsg
18
19    SECTION .bss                 ; Section containing uninitialized data
20
(gdb)
21    SECTION .text                ; Section containing code
22
23    global      _start                 ; Linker needs this to find the entry
point!
24
25    _start:
26          nop              ; This no-op keeps gdb happy...
27          mov eax,4        ; Specify sys_write call
28          mov ebx,1        ; Specify File Descriptor 1: Standard Output
29          mov ecx,EatMsg         ; Pass offset of the message
30          mov edx,EatLen         ; Pass the length of the message
(gdb)
31          int 80H                ; Make kernel call
32
33          MOV eax,1        ; Code for Exit Syscall
34          mov ebx,0        ; Return a code of zero
35          int 80H                ; Make kernel call
36
37
38
39
40
(gdb) break 27
Breakpoint 1 at 0x8048081: file ./eatsyscall.asm, line 27.
```

```
(gdb) list
41
42
43
44
45
46
47
48
49
50
(gdb) display/i $eip
(gdb) run
Starting program: /home/someone/Documents/assembly/eatsyscall/eatsyscall

Breakpoint 1, 0x08048081 in _start ()
1: x/i $eip
=> 0x8048081 <_start+1>:      mov    $0x4,%eax
(gdb) si
0x08048086 in _start ()
1: x/i $eip
=> 0x8048086 <_start+6>:      mov    $0x1,%ebx
(gdb) info registers
eax            0x4      4
ecx            0x0      0
edx            0x0      0
ebx            0x0      0
esp            0xffffd420     0xffffd420
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048086      0x8048086 <_start+6>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) x
0x804808b <_start+11>:  0x0490a4b9
(gdb) si
0x0804808b in _start ()
1: x/i $eip
=> 0x804808b <_start+11>:     mov    $0x80490a4,%ecx
(gdb) info registers
eax            0x4      4
ecx            0x0      0
edx            0x0      0
ebx            0x1      1
esp            0xffffd420     0xffffd420
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x804808b      0x804808b <_start+11>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) x help
No symbol "help" in current context.
```

```
(gdb) help x
Examine memory: x/FMT ADDRESS.
ADDRESS is an expression for the memory address to examine.
FMT is a repeat count followed by a format letter and a size letter.
Format letters are o(octal), x(hex), d(decimal), u(unsigned decimal),
  t(binary), f(float), a(address), i(instruction), c(char), s(string)
  and z(hex, zero padded on the left).
Size letters are b(byte), h(halfword), w(word), g(giant, 8 bytes).
The specified number of objects of the specified size are printed
according to the format.

Defaults for format and size letters are those previously used.
Default count is 1.  Default address is following last thing printed
with this command or "print".
(gdb) si
0x08048090 in _start ()
1: x/i $eip
=> 0x8048090 <_start+16>:    mov    $0xe,%edx
(gdb) info registers
eax            0x4      4
ecx            0x80490a4      134516900
edx            0x0      0
ebx            0x1      1
esp            0xffffd420     0xffffd420
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048090      0x8048090 <_start+16>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) x/10xb
0x8048095 <_start+21>:  0xcd  0x80  0xb8  0x01  0x00  0x00  0x00  0xbb
0x804809d <_start+29>:  0x00  0x00
(gdb) x/10xb 0x80490a4
0x80490a4 <EatMsg>:     0x45  0x61  0x74  0x20  0x61  0x74  0x20  0x4a
0x80490ac:  0x6f  0x65
(gdb) x/10cb 0x80490a4
0x80490a4 <EatMsg>:     69 'E'      97 'a'      116 't'     32 ' '      97 'a'
      116 't'     32 ' '      74 'J'
0x80490ac:  111 'o'     101 'e'
(gdb) where
#0  0x08048090 in _start ()
(gdb) si
0x08048095 in _start ()
1: x/i $eip
=> 0x8048095 <_start+21>:    int    $0x80
(gdb) info registers
eax            0x4      4
ecx            0x80490a4      134516900
edx            0xe      14
ebx            0x1      1
esp            0xffffd420     0xffffd420
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048095      0x8048095 <_start+21>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
```

```
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) x/15is 0x8048095
0x8048095 <_start+21>:  "\270\001"
0x804809a <_start+26>:  ""
0x804809b <_start+27>:  ""
0x804809c <_start+28>:  "\273"
0x804809e <_start+30>:  ""
0x804809f <_start+31>:  ""
0x80480a0 <_start+32>:  ""
0x80480a1 <_start+33>:  ""
0x80480a4:  "Eat at Joe's!\n"
0x80480b3:  ""
0x80480b4:  "\001"
0x80480b6:  ""
0x80480b7:  ""
0x80480b8:  ""
0x80480b9:  ""
(gdb) x/15ib 0x8048095
=> 0x8048095 <_start+21>:     int    $0x80
   0x8048097 <_start+23>:     mov    $0x1,%eax
   0x804809c <_start+28>:     mov    $0x0,%ebx
   0x80480a1 <_start+33>:     int    $0x80
   0x80480a3:     add    %al,0x61(%ebp)
   0x80480a6:     je     0x80480c8
   0x80480a8:     popa
   0x80480a9:     je     0x80480cb
   0x80480ab:     dec    %edx
   0x80480ac:     outsl  %ds:(%esi),(%dx)
   0x80480ad:     gs
   0x80480ae:     daa
   0x80480af:     jae    0x80480d2
   0x80480b1:     or     (%eax),%al
   0x80480b3:     add    %al,(%ecx)
(gdb) si
Eat at Joe's!
0x08048097 in _start ()
1: x/i $eip
=> 0x8048097 <_start+23>:     mov    $0x1,%eax
(gdb) info registers
eax            0xe      14
ecx            0x80490a4      134516900
edx            0xe      14
ebx            0x1      1
esp            0xffffd420     0xffffd420
ebp            0x0      0x0
esi            0x0      0
edi            0x0      0
eip            0x8048097      0x8048097 <_start+23>
eflags         0x202    [ IF ]
cs             0x23     35
ss             0x2b     43
ds             0x2b     43
es             0x2b     43
fs             0x0      0
gs             0x0      0
(gdb) si
0x0804809c in _start ()
1: x/i $eip
=> 0x804809c <_start+28>:     mov    $0x0,%ebx
(gdb) si
0x080480a1 in _start ()
```

```
1: x/i $eip
=> 0x80480a1 <_start+33>:    int    $0x80
(gdb) si
[Inferior 1 (process 3285) exited normally]
(gdb) kill
The program is not being run.
(gdb) quit
someone@debian-box:~/Documents/assembly/eatsyscall$ ./eatsyscall
Eat at Joe's!
someone@debian-box:~/Documents/assembly/eatsyscall$
```