

```

1  zbhuang@zbhuang-VirtualBox:~/OpenJDK/cppapps$ gcc -g functionf.c -o functionf.o
2  zbhuang@zbhuang-VirtualBox:~/OpenJDK/cppapps$ gdb -q functionf.o | tee functionf.asm.s
3  Reading symbols from functionf.o...done.
4  (gdb) disassemble /m main
5  Dump of assembler code for function main:
6      11      {
7          0x080483f6 <+0>:      push    %ebp
8          0x080483f7 <+1>:      mov     %esp,%ebp
9          0x080483f9 <+3>:      sub     $0x10,%esp
10
11      12          int i = 77;
12          0x080483fc <+6>:      movl    $0x4d,-0x4(%ebp)
13
14      13          i = f(i, 8);
15          0x08048403 <+13>:     push    $0x8
16          0x08048405 <+15>:     pushl   -0x4(%ebp)
17          0x08048408 <+18>:     call   0x080483db <f>
18          0x0804840d <+23>:     add     $0x8,%esp
19          0x08048410 <+26>:     mov     %eax,-0x4(%ebp)
20
21      14          i %= 5;
22          0x08048413 <+29>:     mov     -0x4(%ebp),%ecx
23          0x08048416 <+32>:     mov     $0x66666667,%edx
24          0x0804841b <+37>:     mov     %ecx,%eax
25          0x0804841d <+39>:     imul    %edx
26          0x0804841f <+41>:     sar     %edx
27          0x08048421 <+43>:     mov     %ecx,%eax
28          0x08048423 <+45>:     sar     $0x1f,%eax
29          0x08048426 <+48>:     sub     %eax,%edx
30          0x08048428 <+50>:     mov     %edx,%eax
31          0x0804842a <+52>:     shl     $0x2,%eax
32          0x0804842d <+55>:     add     %edx,%eax
33          0x0804842f <+57>:     sub     %eax,%ecx
34          0x08048431 <+59>:     mov     %ecx,%eax
35          0x08048433 <+61>:     mov     %eax,-0x4(%ebp)
36
37      15          return i;
38          0x08048436 <+64>:     mov     -0x4(%ebp),%eax
39
40      16      }
41          0x08048439 <+67>:     leave
42          0x0804843a <+68>:     ret
43
44  End of assembler dump.
45  (gdb) disassemble /m f
46  Dump of assembler code for function f:
47      3      {
48          0x080483db <+0>:      push    %ebp
49          0x080483dc <+1>:      mov     %esp,%ebp
50          0x080483de <+3>:      sub     $0x10,%esp
51
52      4          int i, j;
53      5          i = x;
54          0x080483e1 <+6>:      mov     0x8(%ebp),%eax
55          0x080483e4 <+9>:      mov     %eax,-0x8(%ebp)
56
57      6          j = i * y;
58          0x080483e7 <+12>:     mov     -0x8(%ebp),%eax
59          0x080483ea <+15>:     imul    0xc(%ebp),%eax
60          0x080483ee <+19>:     mov     %eax,-0x4(%ebp)
61
62      7          return j;
63          0x080483f1 <+22>:     mov     -0x4(%ebp),%eax
64
65      8      }
66          0x080483f4 <+25>:     leave
67          0x080483f5 <+26>:     ret
68
69  End of assembler dump.
70  (gdb) run
71  Starting program: /home/zbhuang/OpenJDK/cppapps/functionf.o
72  [Inferior 1 (process 17670) exited with code 01]
73  (gdb) quit
74

```