3.1

|  |  |
| --- | --- |
| 操作数 | 值 |
| %eax | 0x100 |
| 0x104 | 0xAB |
| $0x108 | 0x108 |
| (%eax) | 0xFF |
| 4(%eax) | 0xAB |
| 9(%eax,%edx) | 0x11 |
| 260(%ecx,%edx) | 0x13 |
| 0xFC( ,%ecx,4) | 0xFF |
| (%eax,%edx,4) | 0x11 |

3.5

void decode1(int \*xp, int \*yp, int \*zp)

{

int a = \*yp;

int b = \*zp;

int c = \*xp;

\*yp = c;

\*zp = a;

\*xp = b;

}

3.7

|  |  |  |
| --- | --- | --- |
| 指令 | 目的 | 值 |
| addl %ecx, (%eax) | 0x100 | 0x100 |
| subl %edx, 4(%eax) | 0x104 | 0xA8 |
| imull $16, (%eax,%edx,4) | 0x10C | 0x110 |
| incl 8(%eax) | 0x108 | 0x14 |
| decl %ecx | %ecx | 0x0 |
| subl %edx, %eax | %eax | 0xFD |

3.18

int test(int x, int y)

{

int val = x ^ y;

if (x < -3)

{

if (x > y)

val = x \* y;

else

val = x + y;

}

else if (x > 2)

val = x - y;

return val;

}

3.20

**A.**

|  |  |  |
| --- | --- | --- |
| 寄存器 | 变量 | 初始 |
| %eax | x | x |
| %ecx | y | y |
| %edx | n | n |

**B.**

test-expr: (n > 0) && (y < n), 对应8-11行.

body-statement: x += n; y \*= n; n--; 对应5-7行.

**C.**

1 movl 8(%ebp), %eax %eax = x

2 movl 12(%ebp), %ecx %ecx = y

3 movl 16(%ebp), %edx %edx = n

4 .L2: loop:

5 addl %edx, %eax x += n;

6 imull %edx, %ecx y \*= n;

7 subl $1, %edx n -= 1;

8 testl %edx, %edx test n;

9 jle .L5 if n <= 0, goto exit

10 cmpl %edx, %ecx campare y:n

11 jl .L2 if y < n, goto loop

12 .L5: exit:

3.24

**A.**

会得到:

int sum = 0;;

int i = 0;

while (i < 10)

{

if (i & 1)

continue;

sum += i;

i++;

}

错误:

if为真时无法更新i的值, 导致死循环.

**B.**

int sum = 0;;

int i = 0;

while (i < 10)

{

if (i & 1)

goto L;

sum += i;

L:

i++;

}

3.34

**A.**

参数x的值.

**B.**

int rfun(unsigned x)

{

if (x == 0)

return 0;

unsigned nx = x >> 1;

int rv = rfun(nx);

return (x&1)+rv;

}

**C.**

统计x的二进制表示中1的个数.

3.56

**A.**

分别为%esi, %ebx, %edi和%edx.

**B.**

分别为1431655765和-2147483648.

**C.**

mask != 0

**D.**

mask = ( (unsigned) mask >> n)

**E.**

result ^= (mask & x)

**F.**

int loop (int x, int n)

{

int result = 1431655765;

int mask;

for (mask = -2147483648; mask != 0; mask = ( (unsigned) mask )>> n)

result ^= (mask & x);

return result;

}