ICS 第三章

5. (填表内容): movsbl %al, (%edx)
movb %al, (%edx)
movl %eax, (%edx)
movswl %ax, (%edx)
movzbl %al, (%edx)
movzbl %al, (%edx)
movl %eax, (%edx)

6. (1) 分别为 R[ebp]+8, R[ebp]+12, R[ebp]+16。

(2)
```
void func (int *xptr, int *yptr, int* zptr) {
    int tempx = *xptr;
    int tempy = *yptr;
    int tempz = *zptr;
    *yptr = tempx;
    *zptr = tempy;
    *xptr = tempz;
}
```

11. (1) je 的转移目标地址为 0x804838c + 2 + 0x08 = 0x8048396;
call 的转移目标地址 0x 80483b1 = 0x804838e + 5 + 0x1e, 其中偏移量 1e 在指令中给出, call 指令机器代码占 5 字节, 0x804838e 为当前指令地址。

(2) jb 的转移目标地址为 0x8048390+2+0xf6 = 0x8048488;
movl 指令机器代码共 10 字节, 后 8 字节为两个立即数, 采用小端方式, 故第一个立即数为 0X0804a800, 即 0x804a800, 最后 4 个字节为立即数 00000001H。

(3) mov 地址 x 满足: 0x8049ae0 = x + 0x16, 故 x = 0x8049aca.

(4) jmp 的转移目标地址为 0x804829b + 0xfffff00 = 0x804819b.

15. 缺少部分为

```
4    while (x!=0) {
5        y= y^x ;
6        x >>1 ; }
7    return  y &0x1 ;
```

f1的功能为返回 (x ^ (x>>1) ^ (x>>2) ^ ···) & 0x1，检测x的奇偶性，当x中有奇数个1时返回1；否则返回0；

17. test函数原型为：

unsigned int test(char a, unsigned short b, unsigned short c, short *p);

22. M=5, N=7;

25. (1) node 所需存储空间 4+(4+4)+4=16字节；p, s.x, s.y, next的偏移量为0, 4, 8, 12

(2)
```
np→s.x = np→s.y;
np→p = &(np→s.x);
np→next = np;
```

28. 每个成员偏移量为

c	d	i	s	*p	l	g	*v
0	8	16	20	24	28	32	40

共有48字节，调整顺序为变量长度由大到小：

d	g	i	p	l	v	s	c
0	8	16	20	24	28	32	34

共用34+6=40字节。

31. (1) 添加注释如下：

```
movl   8(%ebp), %edx        // R[edx] ← M[R[ebp]+8] ，将 x 送 edx.
movl   12(%ebp), %ecx       // R[ecx] ← M[R[ebp]+12] ，将 k 送 ecx.
movl   $255. %esi           // R[esi] ← 255
movl   $0x-2147483648, %edi // R[edi] ← -2147483648
.L3:
movl   %edi, %eax           // R[eax] ← R[edi]，将 i 送 eax；
andl   %edx, %eax           // R[eax] ← R[eax] and R[edx] ，i and x 送 eax.
xorl   %eax, %esi           // R[esi] ← R[esi] xor R[eax]，val xor (i and x)送esi
movl   %ecx, %ebx           // R[ebx] ← R[ecx]，k 送 ebx.
shrl   %bl, %edi            // R[edi] ← R[edi] >> R[bl]，i右移k位送edi
testl  %edi, %edi
jne    .L3                  // 若 R[edi]≠0 则转 .L3
movl   %esi, %eax           // R[eax] ← R[esi]
```

(2) x和k在 edx 和 ecx 中；val和i在 esi和edi中；

(3) val和i的初值为 255 和 -2147483648；

(4) 终止循环条件为 i=0，每次循环i逻辑右移k位；

(5) 缺少部分为：

```
int val = 255;
int i;
for (i = -2147483648; i!=0; i = (unsigned) i >> k)
{   val ^= (i & x); }
return val;
```

33. (1) 分别为 0，4，0，4；

(2) node类型共占 8字节；

(3) 缺少部分为：uptr→n2.next→n1.data1 = *(uptr→n2.next→n1.ptr) - uptr→n2.data2;