No.

Date . .

4. (1) 源操作数0x甲是立即数却没有加"$".

(2) 源操作数%ax长度是16却用了后缀"b".

(3) 目的操作数0xF0是立即数.

(4) 源操作数0xFFFF0长度不止16位却用了后缀"w".

(5) 目的操作数地址%dl的长度是8.

(6) 源操作数%bx(8位)和目的操作数%eax(32位)长度不同.

(7) 目的操作数%esx所示寄存器并不存在.

(8) 源操作数8(%ebp.,4)没有反址寄存器.

5. Src_type dst_type 机器级表示

char	int	movsbl %al,(%edx)
int	char	movb %al,(%edx)
int	unsigned	movl %eax,(%edx)
short	int	movswl %ax,(%edx)
unsigned char	unsigned	movzbl %al,(%edx)
char	unsigned	movzbl %al,(%edx)
int	int	movl %eax,(%edx)

6. xptr: R[ebp]+8. yptr: R[ebp]+12. zptr: R[ebp]+16.

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

15. `while (x!=0) { y = y^x; x = x>>1; } return y & 0x1.`

No.

Date . .

函数f1的返回值是(X^X>>1^X>>2^…)&0x1

其中异或操作用于计算X中非零位的个数，

若2进制串X中1的个数为奇数则返回1，为偶数则返回0。

VI. unsigned mb tests(char a, unsigned short b, unsigned short c, short *p);

22. 由此确定数组引用 A[i][j] 位置(地址)为 A[0][0]+7i+j

b[j][i] 的地址为 b[0][0]+5j+i. 即 M=5 N=7.

25. 结构 node 所需存储空间有 4+(4+4)+4=16字节.

成员 p. s.x. s.y 和 next 的偏移地址分别为 0. 4. 8 和 12.

(2) np_Ints 中缺失的表达式分别为：

np->s.x = np->s.y; np->p = &(np->s.x); np->next = np;

28. c d i s p l q v

0 8 16 20 24 28 32 40

结构中d和q按8字节边界对齐，则结构总大小为48字节

将成员按从大到小顺序排列所占空间最小，则调整为

struct { double d; long long q; int i; char* p;

long l; void * v; short s; char c; } test;

每个成员的偏移为：

d q i p l v s c

0 8 16 20 24 28 32 34 结构总大小为40字节.

31. (1) 1. movl 8(%ebp), %edx // 将x送入EDX

2. movl 12(%ebp), %ecx // 将k送入ECX

3. movl $255, %esi // 将255送入ESI

4. movl $-2147483648, %edi // 将 0x80000000 送入EDI

(2) .L3

6. movl %edi, %edx // 将i送入EAX

7. andl %edx, %eax // 将i与x送入EAX

No.

Date . .

```
8   xorl   %eax, %esi        // 将 val ^ (1<<x) 送入 ESI
9   movl   %ecx, %ebx        // 将 k 送入 EBX
10  shrl   %bl, %edi         // 将 1 >> k (逻辑右移) 送入 EDI
11  testl  %edi, %edi
12  jne    .L8               // 若 1!=0, 则跳转到 .L8
13  movl   %esi, %eax        // 将 ESI 中值送入 EAX.
```

(2) 参数 x 和 k 分别存放在寄存器 EDX 和 ECX 中.

局部变量 val 和 i 分别存放在寄存器 ESI 和 EDI 中.

(3) 局部变量 val 和 i 所初始值分别为 0x ff 和 0x80000000.

(4) 循环终止条件是 i==0, 循环控制要每循环一次逻辑右移一位.

(5)
```
int lproc (int x, int k) {
    int val = 255;
    int i;
    for (i = 0x80000000; i!=0; i = unsigned (i >> k))
        val ^= x;
    return val;
}
```

33. (1) n1.ptr, n1.data1, n2.data2, n2.next 每个偏移量分别是 0, 4, 0, 4.

(2) node 类型总大小占 8 字节.

(3) uptr->n2.next->n1.data1 = *(uptr->n2.next->n1.ptr) - uptr->n2.data2;