

计算机系统基础第二次作业

201300035 方盛俊

《计算机系统基础》教材第2版第3章后习题中的第3、4、5、6、7、8、11、13题作业

3.

- (1) 后缀: w; 源: 基址加比例变址加位移; 目的: 寄存器;
- (2) 后缀: b; 源: 寄存器; 目的: 基址加位移;
- (3) 后缀: l; 源: 基址加比例变址; 目的: 寄存器;
- (4) 后缀: b; 源: 基址寻址; 目的: 寄存器;
- (5) 后缀: l; 源: 立即数; 目的: 栈;
- (6) 后缀: l; 源: 立即数; 目的: 寄存器;
- (7) 后缀: w; 源: 寄存器; 目的: 寄存器;
- (8) 后缀: l; 源: 基址加变址加位移; 目的: 寄存器.

4.

- (1) 源操作数应该为立即数 \$0xFF.
- (2) movb 的后缀 b 与 %ax 寄存器长度不符.
- (3) 目的操作数不能是立即数.
- (4) orw 的后缀 w 长度只有 16 位, 无法处理超过 16 位的源立即数 \$0xFFFF0.
- (5) 不能用 8 位寄存器作为操作数地址所在寄存器.
- (6) movl 和后缀 l 与源寄存器 %bx 长度不符.
- (7) 不存在 %esx 寄存器.
- (8) 基址加比例变址加位移寻址, 不能没有变址寄存器.

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.

(1)

xptr, yptr, zptr 对应实参所存放的存储单元分别为: R[ebp] + 8, R[ebp] + 12, R[ebp] + 16.

(2)

```
void func(int *xptr, int * yptr, int *zptr) {  
    int x = *xptr, y = *yptr, z = *zptr;  
    *y = x;  
    *z = y;  
    *x = z;  
}
```

7.

(1) 结果为 `x` .

(2) 结果为 `x + y + 4` .

(3) 结果为 `x + 8 * y` .

(4) 结果为 `y + 2 * x + 12` .

(5) 结果为 `4 * x` .

(6) 结果为 $x + y$.

8.

(1) $R[edx] = 0x0000\ 0070$, $OF = 0$, $SF = 0$, $ZF = 0$, $CF = 1$.

(2) $R[ecx] = 0x8000\ 0008$, $OF = 1$, $SF = 1$, $ZF = 0$, $CF = 0$.

(3) $R[bx] = 0xff00$, $OF = 0$, $SF = 1$, $ZF = 0$, $CF = 0$.

(4) $OF = 0$, $SF = 1$, $ZF = 0$, $CF = 0$.

(5) $R[ecx] = 0x11e2\ 5500$, $OF = 1$, $CF = 1$.

(6) $R[ax] = 0x0000$, $R[dx] = 0x0093$, $OF = 1$, $CF = 1$.

(7) $R[cx] = 0x000f$, $OF = 0$, $SF = 0$, $ZF = 0$.

11.

(1)

je 指令转移目标地址是 $0x804838c + 2 + 8 = 0x8048396$; 因为 $0xb1 - 5 - 0x8e = 0x1e$, 因此目标地址是: $EIP + \text{call 指令长度} + \text{后四个字节组成的偏移量 (小端方式)}$ 得到的.

(2)

jb 指令转移目标地址是 $0x8048390 + 2 - 10 = 0x8048388$; 前两个字节是 opcode, 紧接着后面四个字节为立即数 $0x804a800$, 最后一个字节为立即数 $0x1$.

(3)

mov 指令所在地址是 $0x80492e0 - 0x16 - 2 + 2 = 0x80492ca$.

(4)

jmp 指令转移目标地址是 $0x8048296 + 5 - 0x100 = 0x804819b$.

13.

```
int func(int x, int y) {  
    int z = x * y;  
    if (x <= -100) {  
        if (y > x)  
            z = x + y;  
        else  
            z = x - y;  
    } else if (x >= 16)  
        z = x & y;  
    return z;  
}
```