计算机系统原理

课后作业

要求:

- 请在本周日23:00之前完成。
- 如果遇到在不适合在word中输入完成的题目，可以拍照后粘贴到word文档中。
- 完成后将word文档压缩后上传BB平台，上传的文档格式为：学号_姓名.rar 。

1. 已知函数 comp 的 C 语言代码及其过程体对应的汇编代码如下：

```
1    void comp(char x, int *p)
2    {
3        if (p && x<0)
4            *p += x;
5    }
```

```
1    movb    8(%ebp), %dl
2    movl    12(%ebp), %eax
3    testl   %eax, %eax
4    je      .L1
5    testb   $0x80, %dl
6    je      .L1
7    addb    %dl, (%eax)
8  .L1:
```

请给出每条汇编指令的注释。

```
1    movb    8(%ebp), %dl      //
2    movl    12(%ebp), %eax    //
3    testl   %eax, %eax        //
4    je      .L1               //
5    testb   $0x80, %dl        //
6    je      .L1               //
7    addb    %dl, (%eax)       //
8  .L1:
```

2. 已知函数 do_loop 的 C 语言代码如下：

```
1  short do_loop(short x, short y, short k) {
2  do {
3    x*=(y%k) ;
4    k- - ;
5        } while ((k>0) && (y>k));
6  return x;
7    }
```

函数 do_loop 的过程体对应的汇编代码如下，给每条汇编指令添加注释，并说明每条指令执行后，目的寄存器中存放的是什么信息。

```
1     movw   8(%ebp), %bx
2     movw   12(%ebp), %si
3     movw   16(%ebp), %cx
4   .L1:
5     movw    %si, %dx
6     movw    %dx, %ax
7     sarw    $15, %dx
8     idiv    %cx
9     imulw   %dx, %bx
10    decw    %cx
11    testw   %cx, %cx
12    jle     .L2
13    cmpw    %cx, %si
14    jg      .L1
15  .L2:
16    movswl   %bx, %eax
```

```
1  movw    8(%ebp),  %bx     // R[bx] = M[R[ebp]+8]
2  movw    12(%ebp),  %si    // R[si] = M[R[ebp]+12]
3  movw    16(%ebp),  %cx    // R[cx] = M[R[ebp]+16]
4  .L1:
5  movw      %si,  %dx       // R[dx] = R[si]
6  movw    %dx,  %ax         // R[ax] = R[dx]
7  sarw    $15,  %dx         // 将dx中存放的值向右算术右移15位
8  idiv    %cx               // 16位有符号除法，被除数高位dx，低位ax，被除数
9  imulw    %dx,  %bx        // 有符号乘法，R[bx] = R[dx] * R[bx]
10 decw     %cx              // R[cx] - -
11 testw   %cx,  %cx         // 做逻辑与操作，结果存在eflag中
12 jle       .L2              // 小于等于时跳转(即ZF=1或SF!=OF)
13 cmpw    %cx,  %si         // 将R[cx]与R[si]比较
14 jg        .L1             // 大于时跳转
15  .L2:
16 movswl   %bx,  %eax       // 将bx进行符号扩展成双字后送给eax
```

3. 已知函数f1 的C 语言代码框架及其过程体对应的汇编代码如下，根据对应的汇编代码填写C代码中缺失部分。

1. int f1(unsigned x)
2. {
3. int y = 0 ;
4. while (x!=0) {
5. y = y^x ;
6. x--;
7. }
8. return y;
9. }
	f1:
	pushl %ebp // 将ebp入栈
	movl %esp, %ebp // R[ebp]=R[esp]
	subl $16, %esp // R[esp]=R[esp]-16
	movl $0, -4(%ebp) // M[R[ebp]-4]=0
	jmp .L2
	.L3:
	movl -4(%ebp), %eax // R[eax]=M[R[ebp]-4]
	xorl 8(%ebp), %eax // R[eax]=R[eax]^M[R[ebp]+8]
	movl %eax, -4(%ebp) // M[R[ebp]-4]=R[eax]
	subl $1, 8(%ebp) // M[R[ebp]+8]--
	.L2:
	cmpl $0, 8(%ebp) // 比较0和M[R[ebp]+8]
	jne .L3 不相等则跳转
	movl -4(%ebp), %eax // R[eax]=M[R[ebp]-4]
	leave
	ret

10.	

4. 填表，说明每个数组的元素大小、整个数组的大小以及第i 个元素的地址。64位

数组	元素大小（B）	数组大小（B）	起始地址	元素的地址
char A[10]	1	10	&A[0]	&A[0]+i
Int B[100]	4	400	&B[0]	&B[0]+4i
Short *C[5]	8	40	&C[0]	C[0]+8i
Short **D[6]	8	8	&D[0]	*D[0]+8i
long double E[10]	8	80	&E[0]	&E[0]+8i
long double *F[10]	8	8	&F[0]	F[0]+8i

5. 假设short 型数组S 的首地址AS 和数组下标（索引）变量i（int 型）分别存放在寄存器EDX 和ECX中，下列给出的表达式的结果存放在EAX 或AX中，仿照例子填表，说明表达式的类型、值和相应的汇编代码。

表达式	类型	值	汇编代码
S	short *	AS	leal (%edx), %eax
S+i	short*	*AS*+2**i*	leal (%edx, %ecx, 2), %eax
S[i]	short	M[*AS*+2**i*]	movw (%edx, %ecx, 2), %ax
&S[10]	short *	*AS*+20	leal 20(%edx), %eax
&S[i+2]	short *	AS+2i+4	leal 4(%edx, %ecx, 2), %eax
&S[i]- S	short*	(*AS*+2**i*-*As*)/ 2= *i*	movl %ecx, %eax
S[4*i+4]	short	M[AS+8i+8]	movw 4(%edx, %ecx, 8), %eax
*(S+i- 2)	short	M[AS+2i-4]	movw -2(%edx, %ecx, 2), %eax