

练习题

计 34 董胤蓬 2013011367

(一) 0xFFFFFFFF9B

0xFFFFFFFF05

0x00000031

(二) -121

(三)

A. $5 = (101)_2$, $E = 2$, $M = (1.01)_2 = 5/4$, $f = (0.01)_2 = 1/4$, $V = 5$, 符号位为 0, 指数位为 $100\cdots 01$, 小数位为 $010\cdots 0$

B. $E = n$, $M = (1.11\cdots 1)_2 = 2 - 2^{-n}$, $f = (0.11\cdots 1)_2 = 1 - 2^{-n}$, $V = 2^{n+1} - 1$, 符号位为 0, 指数位为 $n + 2^{k-1} - 1$, 小数位为 $011\cdots 1$

C. $E = 2 - 2^{k-1}$, $M = 1$, $f = 0$, $V = 2^E$, 符号位为 0, 指数位为 $00\cdots 01$, 小数位为 $00\cdots 0$

(四) `int decode2(int x, int y, int z)`

```
{  
    return (((z - y) << 15) >> 15) * (x ^ (z - y))  
}
```

(五) 参数 w, x 通过通用寄存器传递, w 由 %ecx 传递, x 由 %edx 传递, 参数 z, y 通过栈传递, z 先入栈, y 后入栈, 传递参数的栈由被调用者恢复。

(六) `int function(int n)`

```

{
    int res = 0;
    int i, j;
    for(int i = 0; i < 10; i++)
    for(int j = 0; j < 10; j++)
    {
        if((i + j) % 2) res += N[i][j];
        else res -= N[i][j];
    }
}

```

(七)

```

_function1:

pushl %ebp

movl %esp, %ebp

subl $56, %esp

movl _i, %eax

leal -40(%ebp), %edx

movl %edx, (%esp)

movl %eax, 4(%esp)

```

```
call _return_struct
```

```
subl $4, %esp
```

```
xorl %eax, %eax
```

```
leave
```

```
ret
```

```
_return_struct:
```

```
pushl %ebp
```

```
movl %esp, %ebp
```

```
pushl %ebx
```

```
subl $36, %esp
```

```
movl 12(%ebp), %ecx
```

```
movl 8(%ebp), %eax
```

```
leal (%ecx,%ecx), %ebx
```

```
leal (%ecx,%ebx), %edx
```

```
movl %edx, _i
```

```
movl %ecx, (%eax)

movl %ecx, 4(%eax)

movl %ebx, 8(%eax)

movl %ecx, 12(%eax)

movl %ecx, 16(%eax)

addl $36, %esp

popl %ebx

popl %ebp

ret $4
```

(八)

```
_input_struct:

pushl %ebp

movl %esp, %ebp

movl 8(%ebp), %eax

movl 24(%ebp), %edx

popl %ebp
```

```
leal (%edx,%eax,2), %eax
```

```
ret
```

```
_function2:
```

```
pushl %ebp
```

```
movl %esp, %ebp
```

```
subl $60, %esp
```

```
movl _i, %eax
```

```
leal (%eax,%eax), %edx
```

```
movl %eax, (%esp)
```

```
movl %eax, 4(%esp)
```

```
movl %edx, 8(%esp)
```

```
movl %eax, 12(%esp)
```

```
movl %eax, 16(%esp)
```

```
call _input_struct
```

```
leave
```

```
ret
```

(九) SET 函数的返回地址为 GET 函数的返回地址，返回值为 1

A 指令后，%eax 储存为输入参数，即存储当前处理器以及栈信息的内存块地址

B 指令后，%ecx 储存为 GET 函数的 Rtn adr

C 指令，用于储存栈顶地址的上一位地址，在之后 SET 函数中，可以见此地址赋给 %esp

E 指令的作用是将 GET 函数的 Rtn adr 压栈，使 SET 函数返回地址为 GET 函数的返回地址

D 指令为：movl 72(%eax), %esp