# 计算机系统第三章作业

#### 陈俊潼 10185101210

## 3.60

各个变量存储的寄存器如下:

X	n	result	mask
%rdi	%esi	%rax	%rdx

#### 补全后的代码如下:

## 2.63

查看并分析汇编代码:

```
long switch_prob(long x, long n)
                                              > x-48
      x in %rdi, n in %rsi
     0000000000400590 <switch_prob>:
       400590: 48 83 ee 3c
                                                 $0x3c, %rsi
                                         sub
       400594: 48 83 fe 05
 3
                                                $0x5,%rsi
                                         cmp
       400598: 77 29
                                                4005c3 <switch_prob+0x33>
                                         ja
       40059a: ff 24 f5 f8 06 40 00
                                                 *0x4006f8(,%rsi,8)
                                         jmpq
                                                                    x= 48,4
       4005a1: 48 8d 04 fd 00 00 00
7
       4005a8:
                00
       4005a9:
               c3
                                         retq
       4005aa: 48 89 f8
                                                %rdi,%rax
                                         mov
                                                $0x3,%rax
       4005ad:
10
                                         sar
                                                          (x << 4-x) *2+75
       4005b1: c3
11
                                         retq
12
       4005b2:
                48 89 f8
                                         mov
                48 c1 e0 04
13
       4005b5:
                                                $0x4, %rax
                                         shl
       4005b9:
                48 29 f8
                                                %rdi,%rax
14
                48 89 c7
15
       4005bc:
                                                %rax, %rdi
                                         mov
16
       4005bf:
                48 Of af ff
                                                %rdi,%rdi
                                         imul
                                                0x4b(%rdi),%rax
                48 8d 47 4b
17
       4005c3:
                                         lea
       4005c7:
18
                                         retq
```

这里是一个switch语句。第一行会把n减去48,后面的ja为无符号大于,所以减去后的数字也必须大于0。可以推测switch语句是从60开始的(上图截图草稿有误)。不同的跳转点对应的操作都比较简单。整理后补全后的switch语句如下:

```
long switch_prob(long x, long n) {
        long result = x;
        switch (n) {
        case 60:
        case 62:
                 result = x * 8;
                break;
        case 63:
                 result = x \gg 3;
                break;
        case 64:
                 x = x << 4 - x;
        case 65:
                x = x * x;
        default:
                 result = x + 0x4B;
        return result;
}
```

## 3.65

查看并分析汇编代码,得到答案如下:

```
1
   .L6:
2
    movq
3
    movq
           %rsi, (%rdx)
4
    movq
5
           %rcx, (%rax)
    movq
           $8 %rdx
6
    addq
           $120, %rax
7
    addq
                        - ALi+1][i],120/8=15.
8
           %rdi, %rax
    cmpq
9
    jne
           .L6
   我们可以看到 GCC 把数组索引转换成了指针代码。
                          A[i][j]的指针? —> %rdx
A. 哪个寄存器保存着指向数组元
B. 哪个寄存器保存着指向数组元素 A[j][i]的指针? -> 7orax
C. M 的值是多少? Mこバ.
```

# 3.69

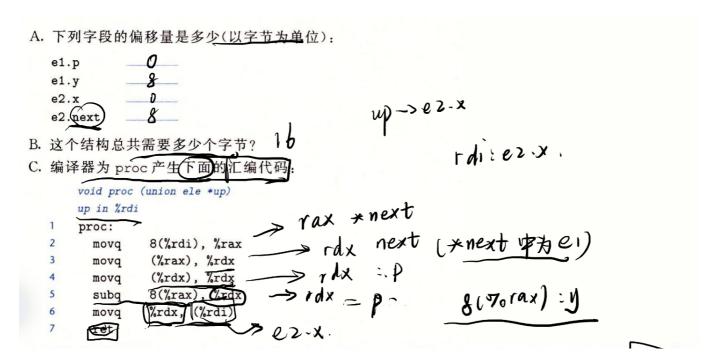
首句中0x120=288, b\_struct内的第一个int和第二个int相差了288个字节。

又从后面的语句mov %rcx, 0x10(%rax,%rdx,8)中, 0x10=40可以推断出每个a\_struct的大小为40, 40\*7+8=288, 所以**CNT=7**。

mov %rcx, 0x10(%rax,%rdx,8)可以推断a\_struct中有一个long类型的x数组和long类型的idx。结合每个a\_struct的大小得到完整定义如下:

```
typedef struct a_struct{
  long idx,
  long x[4]
}
```

### 3.70



分析汇编语句,up中应该为e2类型,其中的next指向的是e1类型。

注意到前面对%rdx连续解了两次引用,关键的subq 8(%rax), %rdx语句后面的8(%rax)存储的是next里的y, %rdx指向的是next里的p指向的long数据。

#### 补全后的语句如下:

```
void proc (union ele * up){
    up->e2.x = *(*(up->e2.next).e1.p) - *(up->e2.next).e1.y;
}
```

# 3.46

选做题好难,我好累Orz.