

计算机系统概论 (2022 秋) 作业 2

1. 使用不超过 3 条 x86 指令实现如下函数：其中  $x, y, z, w$  分别存储于 `%edi`, `%esi`, `%edx`, `%ecx`，返回值存储于 `%eax`。

```
long add(long x, long y, long z, long w) {
    return 32 * x + 8 * y + 4 * z + w;
}
```

add.

$$\frac{\log(\%r_{ii}, \%r_{di}, \%r_{ei})}{\log(\%r_{ii}, \%r_{di}, \%r_{ei}, \%r_{oi})} \cdot \%r_{oi}$$
$$\frac{\log(\%r_{oi}, \%r_{oi}, \%r_{oi})}{\log(\%r_{oi}, \%r_{oi}, \%r_{oi}, \%r_{oi})} \cdot \%r_{oi}$$

ret

2. X86-64 体系结构中的条件跳转指令 `jg` 是用于符号数比较还是无符号数比较的? 其产生跳转的成立条件是 `!(SF^OF)&!ZF` 为真, 请解释为何是这一条件。

有符号数；若  $a > b$ ，则  $a - b > 0$ ，若  $a - b$  未溢出，则  $SF$  为 0 ( $a - b$  为正数)、 $OF$  为 0、 $ZF$  为 0， $\sim(SF \wedge OF) \& \sim ZF = 1$ ；若  $a - b$  溢出，则  $SF$  为 1 ( $a - b$  过大，溢出为负)， $OF$  为 1， $ZF$  为 0， $\sim(SF \wedge OF) \& \sim ZF = 1$

3. 有如下对应的 C 代码与汇编代码 (x86-64), 请对照着填上代码中缺失的部分 (数字请用十进制表示)

**call\_swap:**

```
subq    $24, %rsp
movl    ①, 12(%rsp)
movl    $91125, 8(%rsp)
leaq    8(%rsp), %rsi
leaq    12(%rsp), ④
movl    $0, %eax
```

call      swap

- ① : \$ 15213
- ② : 91125
- ③ : swap (& zip 1, & zip 2)
- ④ : % rdi

```
void swap(int *a, int *b);
void call_swap()
{
    int zip1 = 15213;
    int zip2 = ②;
    ③;
}
```



4. 一个C语言的for循环代码(部分)及其

64位Linux汇编如下所示,请对照汇编填

充C语言里的缺失部分。

```
int loopier(int n, int *a) {
    int i;
    int x = 0;
    for(i = 0;
        i < n;
        i++)
    {
        if (a[i] > x)
            x = 2 * a[i];
        else continue;
    }
    return x;
}
```

```
loopier:
    movl    $0, %eax
    movl    $0, %edx
    jmp     .L2

.L4:
    movslq  %edx, %rcx
    movl    (%rsi,%rcx,4), %ecx
    addl    $1, %eax
    cmpl    %eax, %ecx
    jle     .L3
    leal    (%rcx,%rcx), %eax

.L3:
    addl    $1, %edx

.L2:
    cmpl    %edi, %edx
    jl      .L4
    ret
```

5. 对于如下代码

```
long v2permute(long *array, long x, long y, long z), {
    long t1 = 8253 * x;
    long t2 = array[t1 + 2 * y];
    long t3 = array[t2 * 16 + z];
    long t4 = t1 + t2 + t3;
    long t5 = array[0] * t4;
    long ret = t3 & t5;
    return ret;
}
```

$p[i] = \text{array}[t1 + 2 * y]$   
 $\Rightarrow * (p + i)$   
 $ret \Rightarrow t3 \& t5$

对应如下汇编指令,请写出每条指令之后目标寄存器存储的变量/临时变量值

```
v2permute:
    movq    %rdx, %r8
    movq    %rcx, %r9
    imulq   $8253, %rsi, %rax
    leaq    (%rax,%r8,2), %rcx
    movq    (%rdi,%rcx,8), %rcx
    salq    $4, %rcx
    addq    %rdx, %rcx
    imulq   (%rdi), %rax
    andq    (%rdi,%rcx,8), %rax
    ret
```

y
z
t1
2 * y + t1
t2
t2 * 16
t2 * 16 + z
t5
ret

6. 请对照下面的C语言代码与相应汇编(Linux X86-64),给出M、N的值。

copy\_element:





```

movslq %rsi, %rsi
leaq    (%rsi,%rsi,2), %rax 3j
leaq    (%rsi,%rax,4), %rax 13j
addq    %rdi, %rax i + 13j
movl    mat2(,%rax,4), %edx 2 + 4(i + 13j)
leaq    0(%rdi,8), %rax
subq    %rdi, %rax 7i
addq    %rax, %rsi 7i + j
movl    %edx, mat1(,%rsi,4)
ret

```

```

#define M 13
#define N 7
int mat1[M][N];
int mat2[N][M];
int copy_element(int i, int j)
{
    mat1[i][j] = mat2[j][i];
}

```

