秘密★启用前

2021秋ICS小班班练习题3
建议用时：50分钟

姓名：　　学号：

| Machine Prog: Basics |
| --- |
| Machine Prog: Control |
| Machine Prog: Procedures |

答卷说明：
a.答卷前,考生务必将自己的姓名填写在答题卡指定位置.
b.答题时,请将答案填写在试卷和答题卡相应位置.如需改动,请用签字笔将原答案划去,再在规定位置填写修正后的答案.未在规定区域作答的答案无效.
c.本卷共4页,卷面分110分.考试结束后,试卷由助教统一收回.

一、选择题(50分)　每题只有一个正确答案

约定：如无特别说明,假设代码全部在Intel x86-64上运行.

(　　)1.某C语言程序中对数组变量a的声明为"long a[10][10];",有如下一段代码：
for (i=0; i<10; i++) for (j=0; j<10; j++) sum+= a[i][j];
假设执行到"sum+= a[i][j];"时,sum的值在%rax中,a[i][0]所在的地址在%rdx中,j在%rsi中,则"sum+= a[i][j];"所对应的指令是( ).

A. addl 0(%rdx, %rsi, 8), %eax
B. addl 0(%rsi, %rdx, 8) , %eax
C. addl 0(%rdx, %rsi, 4) , %eax
D. addl 0(%rsi, %rdx, 4) , %eax

(　　)2.下列关于数据传送指令的说法中,正确的是____.

A.常规的movq指令只能以表示为32位补码数字的立即数作为源操作数(注：这里的"只能"强调的是不能以64位数作为源操作数),然后把这个值零扩展到64位的值,放到目的位置.movabsq指令能够以任意64位立即数值作为源操作数,并且只能以寄存器为目的.

B.在Imm(rb,ri,s)这一寻址模式中,s必须是1,2,4或者8,基址和变址寄存器必须是64位寄存器.

C.cqto指令不需要额外操作数,它的作用是把%eax符号扩展到%rax.

D.CPU执行指令"movl -1 %eax"后,%rax的值为0x00000000ffffffff.

(　　)3.下列关于通用目的寄存器和条件码的说法中,正确的是____.

A.%rbx,%rbp,%r12-r15是被调用者保存寄存器,其他寄存器是调用者保存寄存器.

B.xorq %rax %rax可以用于清空%rax;cmpq %rax,%rax可以用于判断%rax的值是否为零.

C.调用一个过程时,最多可以传递6个整型参数,依次存储在(假设参数都是64位的)%rdi,%rsi,%rdx,%rcx,%r8,%r9中.当需要更多的参数时,调用者过程可以在自己的栈帧里按照地址由低到高的顺序依次存储这些参数.

D.leaq指令不改变条件码;逻辑操作(如XOR)会将进位标志和溢出标志都设置为零;移位操作会把溢出标志设置为最后一个被移出(shift out)的位,而把进位标志设置为零.

(　　)4. 在下面的代码中,A和B是用#define定义的常数：

```
typedef struct {int x[A][B]; long y;} str1;
typedef struct {char array[B]; int t; short s[A]; long u;} str2;
void setVal(str1 *p, str2 *q) {
        long v1 = q->t;long v2 = q->u;
        p->y = v1+v2;
}
```

GCC为setVal产生下面的代码:

```
setVal:
movslq 8(%rsi), %rax
addq 32(%rsi), %rax
movq %rax, 184(%rdi)
ret
```

则A=____,B=____.

A. 8,6　　B.10,8　　C.10,5　　D. 9.5

(　　)5.假设某条 C 语言 switch 语句编译后产生了如下的汇编代码及跳转表:

```
movl 8(%ebp), %eax          .L7:
subl $48, %eax              .long .L3
cmpl $8, %eax               .long .L2
ja .L2                      .long .L2
jmp *.L7(, %eax, 4)         .long .L5
                            .long .L4
                            .long .L5
                            .long .L6
                            .long .L2
                            .long .L3
```

在源程序中,下面的哪些(个)标号出现过____.

A. '2', '7'　B. 1　　C. '3'　　D. 5

(　　)6.x86 体系结构中,下面哪个说法是正确的?

A. leal 指令只能够用来计算内存地址

B. x86_64 机器可以使用栈来给函数传递参数

C. 在一个函数内,改变任一寄存器的值之前必须先将其原始数据保存在栈内

D. 判断两个寄存器中值大小关系,只需要 SF 和 ZF 两个条件码

(　　)7.下列的指令组中,哪一组指令只改变条件码,而不改变寄存器的值?

A. CMP, SUB　　B. TEST, AND　　C. CMP, TEST　　D. LEAL, CMP

(　　)8.在下列关于条件传送的说法中,正确的是:

A. 条件传送可以用来传送字节、字、双字、和 4 字的数据

B. C 语言中的"?:"条件表达式都可以编译成条件传送

C. 使用条件传送总可以提高代码的执行效率

D. 条件传送指令不需要用后缀(例如 b, w, l, q)来表明操作数的长度

(　　)9.以下关于 x86-64 指令的描述,说法正确的有几项?

(a) 有符号除法指令 idivq S 将%rdx(高 64 位)和%rax(低 64 位)中的128 位数作为被除数,将操作数 S 的值作为除数,做有符号除法运算;指令将商存在%rdx 寄存器中,将余数存在%rax 寄存器中.(b) 我们可以使用指令 jmp %rax 进行间接跳转,跳转的目标地址由寄存器%rax 的值给出.(c) 算术右移指令 shr 的移位量既可以是一个立即数,也可以存放在单字节寄存器%cl 中.(d) leaq 指令不会改变任何条件码.

A. 1　　B. 2　　C. 3　　D. 4

(　　)10.X86-64 指令提供了一组条件码寄存器;其中 ZF 为零标志,ZF=1 表示最近的操作得出的结构为 0;SF 为符号标志,SF=1 表示最近的操作得出的结果为负数;OF 为溢出标志,OF=1 表示最近的操作导致一个补码溢出(正溢出或负溢出).当我们在一条 cmpq 指令后使用条件跳转指令 jg 时,那么发生跳转等价

于以下哪一个表达式的结果为 1?

A. ~(SF ^ OF) & ~ZF　　B. ~(SF ^ OF)　　C. SF ^ OF　　D.(SF^OF)|ZF

二、非选择题(60分) 请将答案填写在答题卡上

11(50分). 以下提供了一段代码的汇编,和语言的情况,请你将横线填写完整.

附:一些可能用到的字符的ASCII码表

| 换行 | 空格 | " | % | ( | ) | , | 0 | A | a |
|---|---|---|---|---|---|---|---|---|---|
| 0x0a | 0x20 | 0x22 | 0x25 | 0x28 | 0x29 | 0x2c | 0x30 | 0x41 | 0x61 |

汇编:

```
0000000000400596 <func>:
400596: sub $0x28,%rsp
40059a: mov %fs:0x28,%rax
4005a3: mov %rax,0x18(%rsp)
4005a8: xor %eax,%eax
4005aa: mov (%rdi),%rax
4005ad: mov 0x8(%rdi),%rdx
4005b1: cmp %rdx,%rax
4005b4: jge (1)____
4005b6: mov %rdx,(%rdi)
4005b9: mov %rax,0x8(%rdi)
4005bd: mov 0x8(%rdi),%rax
4005c1: test %rax,%rax
4005c4: jne 4005cb <func+0x35>
4005c6: mov (%rdi),%rax
4005c9: jmp (2)____
4005cb: mov (%rdi),%rdx
4005ce: sub %rax,%rdx
4005d1: mov %rdx,(%rsp)
4005d5: mov %rax,0x8(%rsp)
4005da: mov (3)____ ,%rdi
4005dd: callq 400596 <func>
4005e2: mov 0x18(%rsp),%rcx
4005e7: xor %fs:0x28,%rcx
4005f0: (4)____ 4005f7 <func+0x61>
4005f2: callq 400460 <__stack_chk_fail@plt>
4005f7: add (5)____,%rsp
4005fb: retq
00000000004005fc <main>:
4005fc: sub $0x28,%rsp
400600: mov %fs:0x28,%rax
400609: mov %rax,0x18(%rsp)
40060e: xor %eax,%eax
400610: movq 0x69,(%rsp)
400618: movq 0xfc,0x8(%rsp)
400621: mov %rsp,%rdi
400624: callq 400596 <func>
400629: mov %rax,%rsi
40062c: mov $0x4006e4,%edi
400631: mov $0x0,%eax
400636: callq 400470 <printf@plt>
40063b: mov 0x18(%rsp),%rdx
400640: xor %fs:0x28,%rdx
400649: (4)____ 400650 <main+0x54>
40064b: callq 400460
<__stack_chk_fail@plt>
400650: mov $0x0,%eax
400655: add (5)____ ,%rsp
400659: retq
```

C语言:

```
typedef struct{ long a; long b; } pair_type;
long func(pair_type *p) {
        if (p -> a < p -> b) {
                long temp = p -> a;
                p -> a = p -> b;
                p -> b = temp;
        }
        if ((6)____ ) return p -> a;
        pair_type np;
        np.a = (7) ____;  np.b = (8) ____;
        return func(&np);
}
int main(int argc, char* argv[]) {
pair_type np;
np.a = (9) ____; np.b = (10) ____ ;
printf("%ld", func(&np));
return 0;
}
```

| |
|---|
| (1)____ |
| (2)____ |
| (3)____ |
| (4)____ |
| (5)____ |
| (6)____ |
| (7)____ |
| (8)____ |
| (9)____ |
| (10)____ |

12.(10分)已知函数Foo()的汇编代码如下,按要求回答问题.

```
Foo(char*, int):
        pushq   %rbp
        movq    %rsp, %rbp
        movq    %rdi, -24(%rbp)
        movl    %esi, -28(%rbp)
        movl    $0, -4(%rbp)
        subl    $1, -28(%rbp)
        jmp     .L2
.L3:
        movl    -4(%rbp), %eax
        movslq  %eax, %rdx
        movq    -24(%rbp), %rax
        addq    %rdx, %rax
        movzbl  (%rax), %eax
        movb    %al, -5(%rbp)
        movl    -28(%rbp), %eax
        movslq  %eax, %rdx
        movq    -24(%rbp), %rax
        addq    %rdx, %rax
        movzbl  (%rax), %edx
        movq    -24(%rbp), %rsi
        movl    -4(%rbp), %eax
        leal    1(%rax), %ecx
        movl    %ecx, -4(%rbp)
        cltq
        addq    %rsi, %rax
        movb    %dl, (%rax)
        movzbl  -5(%rbp), %edx
        movq    -24(%rbp), %rsi
        movl    -28(%rbp), %eax
        leal    -1(%rax), %ecx
        movl    %ecx, -28(%rbp)
        cltq
        addq    %rsi, %rax
        movb    %dl, (%rax)
.L2:
        movl    -4(%rbp), %eax
        cmpl    -28(%rbp), %eax
        jl      .L3
        nop
        nop
        popq    %rbp
        ret
```

(1).已知Foo()函数的两个参数分别是一个字符数组的首地址和该字符数组的长度.请根据汇编代码简述Foo()函数的功能____(第(2)问C代码调用了Foo()函数，可能对解题有帮助).

(2). 已知Bar()函数和main()函数的C代码如下,则该段程序的输出是____.

```
void Bar( char* buffer, int n, int k ){
   k %= n ;
   Foo( buffer, n-k) ;
   Foo( buffer+n-k, k) ;
   Foo( buffer, n) ;
}
```

```
int main(){
   char buffer[100]="I will change if U change ";
   Bar(buffer,strlen(a),12);
   printf("%s\n",buffer);
   return 0;
}
```

(　　)13.(补充题,不计分 命题人:王星博)下列说法中正确的是____.

A. pushq指令的功能把数据压入到栈上，pushq S 的意义是将寄存器S中的数据保存到寄存器%rsp中

B. 形如 $Imm(r_b, r_i, s)$的寻址格式为比例变址寻址，例如$10(\%eax, \%ebx, 4)$

C. cltq指令是符号扩展指令，不需要操作数

D. 将一个四字值弹出栈后需要将栈指针加8，因此，弹出一个四字的操作包括 $popq\ \%rax; addq\ \$8, \%rsp.$

(　　)14.(补充题,不计分 命题人:王星博)关于汇编指令的说法，正确的是____.

A.算术操作如addq, subq, imulq等都是二元操作，必须有两个操作数

B.指令 $SHL\ \ k, D$ 的效果相当于 $D \leftarrow D \ll k$，指令 $SHL\ \ D$ 是非法的

C.移位操作的移位量可以是立即数，也可以放在单字节寄存器$\%bl$或$\%cl$中，但是不能从内存中读取

D.当强制类型转换既涉及大小变化又涉及C语言中符号变化时，应该先改变大小

*答案：13.C(A: pushq S的意义是将栈指针移动后，将寄存器S中的数据保存到寄存器%rsp指向的内存地址 B: 比例变址寻址的基址和变址寄存器都必须是64位寄存器 C: 正确 D: popq 包含了栈指针加8的操作，弹出一个四字的操作等同于* $movq\ (\%rsp), \%rax; addq\ \$8, \%rsp.$

*14.D (A: imulq可以只有一个操作数 B: shl D 代表将D左移1位 C: 移位量只能存放在单字节寄存器$\%cl$ 中 D: 书中P126原话)*

# 2021秋ICS小班18班第(3)次考试

## 参考答案

1.A
2.B(A符号扩展,C是cltq,D缺少$符)
3.C(A%rsp是例外,B可用于判断是否为零的是test,D移位操作设置进位条件码,但把溢出条件码设置为零)
4.D(4 < B <= 8,5 < A <= 10,44 < A*B <= 46,解得 A=9 B=5)
5.C
解析：‘0’的ASCII码为48，‘1’‘2’‘7’在跳转表中未出现
6.B
解析：A.leal 指令做普通算术运算；C.caller saved 寄存器才需要；D.还需要 OF
7.C
8.D
解析：A.条件传送不支持单字节传送；B.如果“?:”涉及到的两个表达式中有一个出错或者有副作用，用条件传送会导致非法行为；C.如果被旁路的分支的计算量很大，计算就白做了；D.从目标寄存器的名字可以推断出条件传送指令的操作数长度
9.A
本题考察 x86-64 中的一些基本指令，答案为 A。a 项错误，原因是 idivq 将余数存在%rdx 中，将商存在%rax 里。b 项错误，间接跳转的正确书写格式应为 jmp *%rax。c 项错误，算术右移指令应为 sar。
10.A。
解析：本题考察 x86-64 条件码，答案为 A。cmpq a,b 相当于通过b－a的值来设置条件码。SF ^ OF 为 1 表示 b < a（减法结果要么负溢出要么为负数），于是~(SF ^ OF)表示 b >= a，再与上 b != a 的条件（~ZF），就可以得到最终结果(b > a)。
11.略，见往年题
12.　1.数组逆序　2. if U change I will change