

1. x86-64 的程序员可见状态

- (1) 通用寄存器: ____, ____, ____, ____, %rbp, %rsp, ____, ____, %r8~%r16
- (2) 程序计数器(PC): ____
- (3) 条件码: ____, ____, SF, OF
- (4) 其他寄存器(浮点、段等等, 不重要)
- (5) 内存

2. 寄存器的作用

被调用者保存 callee saved	调用者保存 caller saved
参数(第一个到第六个)	其他

3. 在 C 语言中, 定义 `int x = 2;` 判断表达式 "`x & 2 == 2`" 的值: _____

4. () 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`.

5. () 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. `LEA, CMP`

6. () 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

7. () 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
8. () 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(rb, ri, 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`.
9. () 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. $\sim(SF \wedge OF) \& \sim ZF$
 - B. $\sim(SF \wedge OF)$
 - C. $SF \wedge OF$
 - D. $(SF \wedge OF) | ZF$
10. () 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

以下为 21 年期中第二大题

得分

第二题 请结合教材第二章“信息的表示和处理”的相关知识,回答下列问题(10分)

1. 假设某 x86-64 机器在地址 0x100 和 0x101 处存储的数据用二进制表示分别为 $[1010\ 1100]_2$ 和 $[1111\ 1011]_2$. 又假设 x 是一个 short 类型的变量, 其地址为 0x100. 则 x 的十六进制补码表示为 (1) 0x_____, 这是一个 (2) _____ (填“正”或“负”) 数, 其绝对值为 (3) _____ (用十进制数字表示).

2. 设整型变量采用 32 位补码表示, 判断正误 (填“√”或“×”):

i. 设 x, y, z 是整型变量, 且 $x < y < z < 0$, 则 $(-y) > (-z) > 0$. ((4) _____)

ii. 表达式 “ $(-5) + \text{sizeof}(\text{int}) < 0$ ” 为真. ((5) _____)

3. 考虑一种新的遵从 IEEE 规范的浮点格式系统, 该浮点数系统包含 1 位符号, 4 位阶码, 7 位尾数.

i. 该浮点数系统所能表示的最大规格化数 (十进制) 为 (6) _____.

ii. 将 $-\frac{3}{256}$ 用该浮点数系统表示, 写成十六进制为 (7) 0x_____. 按照向偶数

舍入的原则, 把 $\frac{3}{256}$ 的二进制表示舍入到最近的一百二十八分之一, 将得到

(8) _____ (填最简真分数).

4. 有下面一段程序:

```
int i, j, k;
for(i = 0; i <= 2147483647 - 2; i ++){
    j = i + 1;
    k = j + 1;
    float *x = (float *)(&i);
    float *y = (float *)(&j);
    float *z = (float *)(&k);
    if(*y - *x != *z - *y){
        printf("0x%08x\n", i);    //输出 8 位 16 进制数
        break;
    }
}
```

其中 float 类型表示 IEEE-754 规定的浮点数, 包括 1 位符号, 8 位阶码, 23 位尾数. 请问该程序是否会有输出? (9) _____ (填 “是” 或 “否”). 若有输出, 请给出输出内容, 若没有输出, 请说明理由 (10) _____.