**《计算机系统基础》期中试卷参考解答**

**班级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 姓名\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1、在对某程序进行调试时，看到如下一段信息：

004116DA 83 7D FC 00 cmpl ………………

004116DE 75 08  jne 004116E8

004116E0 8B 45 FC mov ………………

004116E3 89 45 F8 mov ………………

004116E6 EB 07 jmp 004116EF

004116E8 C7 45 F8 01 00 00 00 movl ………………

004116EF E8 2D FB FF FF callq func (00411221) ; func为函数名

004116F4 C7 45 F4 21 12 41 00 movl $0x411221, -0x0c(%ebp)

004116FB FF 55 F4 callq -0x0c(%ebp)

………… ………… …………

① 观察指令在内存中的存放形式(每个空对应一个16进制字节数据)：

004116DA \_83\_\_ \_\_7D\_\_ \_FC\_\_ \_00\_\_ \_75\_\_ \_08\_\_ \_\_8B\_\_\_ \_\_45\_\_\_

004116E2 ………………………………

② 设当前 eip 为 0x004116DE，在取出eip指向的指令并进行译码后，eip = \_\_\_0x004116E0\_\_\_\_；

③ 在0x004116DE 处的指令为： 75 08 jne 004116E8，直观上是在zf =\_\_0\_\_\_时，会将0x004116E8 →eip , 计算出该地址值的方法 0x004116E0(下一条指令的地址) +08 ；当转移条件不成立时，则该指令执行完成， eip 不会 （会、不会）改变。

④ 004116EF 处的指令中有0xFFFFFB2D，计算出该值的方法是 00411221 - 004116F4 ；

⑤ 004116FB 处的子程序调用指令，调用的子程序的入口地址是 0x411221 ，执行该处语句时，CPU会将 0x 004116FE 压入堆栈中。

⑥ 执行子程序中的RET指令时，CPU会 从栈顶弹出一个双字 →eip。指令call \*-0x0c(%ebp)中-0x0c(%ebp)的寻址方式为\_\_\_变址寻址\_\_\_\_\_\_\_\_\_\_\_。

2. (1) 设A1，B1，C1均为数据段中定义的有符号字类型变量，用两种方法计算(A1\*5+B1)/8的值，并保存在C1中，编写程序段(不必写出完整的程序)，完成上述功能，不考虑溢出。

方法1： 方法2：

MOV A1, %AX MOV A1，%AX

IMUL %5，%AX SAL $2, %AX

ADD B1，%AX ADD A1, %AX

CWD (漏掉符号扩展扣3分) ADD B1, %AX

MOV $8, %BX SAR $3, %AX

IDIV %BX MOV %AX， C

MOV %AX, C1 注意乘以5还可以用LEA (%EAX, %EAX, 4), %EAX，

也可以用LEA (%EAX, 4), %EAX ADD A1, %AX

只要是方法不一样，指令序列写正确都可以给分，但每类指令只能用一次，即乘除指令要用上，移位或者lea指令要用，但是不可以两种方法都用了SAR或者两种方法都用了乘法指令。

(2) 变量x定义： x: word 10, 20，按下面要求交换x和x+2的值

①只用mov指令 ②只用循环移位指令 ③只用push和pop指令

Mov x, %ax pushw x

Mov x+2, %bx rorl $16, %x （用roll也可以） pushw x+2

Mov %bx, x popw x

Mov %ax, x+2 popw x+2

3. 阅读下面的程序，回答问题。

.section .data

array: .ascii "12345abcAFgG09874\0"

len = . -array # length 为array中元素的个数

num: .fill 10, 1, 0

.section .text

.global \_start

\_start:

mov $len, %ecx ----①

mov $array, %ebx ----②

xor %eax, %eax

L2: movb (%ebx), %al

cmpb $'0', %al

jb L1

cmp $'9', %al

ja L1

sub $0x30, %al

incb num(,%eax,1) ----④

L1: incl %ebx

loop L2

mov $1, %eax ③# 程序正常退出

mov $0, %ebx

int $0x80

1. 该程序的功能是什么？写出程序运行到③时， num所在存储区的10个字节的内容。

统计字符串中每个数字字符（'0' 到 '9'）的出现次数，将结果存储在num数组中。num数组的索引i表示所统计的数字字符，值num[i]表示数字i出现的次数。

[1, 1, 1, 1, 2, 1, 0, 1, 1, 1]

1. 若标号L2写在了②的位置，程序运行结果会怎样？

由于每次循环ebx寄存器总是重新被赋值为array的首地址，每次循环取出的数据值都是第一个字符’1’，因此该程序是将第一个字符重复统计了17次(字符串长度次数).

1. 若指令①写成了mov $0, %ecx，则程序运行会怎样？

若ecx初始被赋值为0，则第一次循环执行loop指令时，会将ecx转换为0xffffffff，从而使得循环不断执行，直到程序运行出现保护性错误（地址越界）。

1. 指出④处的指令的寻址方式

变址寻址