

考试中心填写:

____年 ____月 ____日
考试用

湖南大学课程考试试卷

课程名称: 计算机系统(2019 春); 试卷编号: A; 考试时间: 120 分钟

题号	一	二	三	四	五	六	七	八	九	十	总分
应得分	10	10	30	15	15	20					100
实得分											
评卷人											评分:

一. 填空题 (10 分, 每空 2 分)

1. 0xb1e56f07 存放在采用小端存储的机器上, 地址为 0x3287 到 0x328a, 则 0x3288 处存放值为_____(以十六进制小写格式表示例 0xff)。

2. 根据操作数特点, 用恰当的 MOV 类指令补全下列残缺数据传送指令: _____ %eax,%bh。

3. 已知

%eax=0X100,%ebx=0X3;

内存中指定地址的值列表如下:

地址 值

0X100 0x9c

0X104 0xf4

0X108 0xc5

0X10C 0x5a

则指令 addl 0x40,(%eax,%ebx,4)将地址_____中的值更新为_____。

4. 假设寄存器 %ebx 的值为 0xa3391eb1, 执行以下指令后

```
movl    %ebx,%ecx
```

```
movw    $0x2311,%bx
```

```
movb    $0x03,%bl
```

```
cmpl    %ecx,%ebx
```

```
jae     .L1
```

```
addl    $0x8,%ebx
```

```
jmp     .L2
```

```
.L1:
```

```
subl    $0x9,%ebx
```

```
.L2:
```

%ebx 的值为_____(以十六进制格式小写表示例 0xffffffff)。

二. (10 分)

假设一个基于 IEEE 浮点格式的 9 位浮点表示, 有 1 个符号位, 4 个阶码位 ($k=4$) 和 4 个尾数位 ($n=4$)。

1. 请给出值-1.375 的二进制位表示 (2 分)。
2. 请简单说明规格化值、非规格化值及特殊值的判断方式与值计算方法 (4 分)。
3. 请写出正数中最小的非规格化数、最大的非规格化数、最小的规格化数、最大的规格化数的二进制位表示 (4 分)。

三. (30 分) 以下有两段完整或者不完整的 C 程序段及相应的汇编代码 (在 32 位环境下), 请回答相关问题。

(1) (15 分)

考虑以下代码, 其中 A 和 B 是用#define 声明的常数:

```
int array1[A][B];
```

```
int array2[B][A];
```

```
int test(int I, int j){  
    return array1[i][j] + array2[j][i];  
}
```

编译上述代码得到如下汇编代码:

```
movl    8(%ebp), %ecx  
movl    12(%ebp), %edx  
leal    0(, %ecx, 4), %eax  
subl    %ecx, %eax  
addl    %edx, %eax  
leal    (%edx, %edx, 4), %edx  
addl    %ecx, %edx  
movl    array1(, %eax, 4), %eax  
addl    array2(, %edx, 4), %eax
```

假设 i 在 %ebp + 8 的位置, j 在 %ebp + 12 的位置, 考虑到行优先访问策略, 请根据这段汇编代码确定 A 和 B 的值, 并给出分析过程。

(2) (15 分)

下面的 C 语言代码省略了 switch 语句的主体部分。在 C 代码中, 标号是不连续的, 并且有些情况还有多个标号。

```
int switch(int x){  
    int result = 0;  
    switch(x){  
        /*Some switch body here*/  
    }  
    return result;  
}
```

GCC 编译后生成如下代码。（变量 x 开始时位于相对于寄存器&ebp 偏移量为 8 的地方）

```
movl    8(%ebp), %eax
/*set up jump table access*/
addl    $3, %eax
cmpl    $6, %eax
ja      .L2
jmp     *.L8( , %eax, 4)
```

跳转表如下：

```
.L8:
      .long    .L3
      .long    .L2
      .long    .L4
      .long    .L5
      .long    .L6
      .long    .L6
      .long    .L7
```

根据上述信息回答问题（需要给出详细分析过程）：

A. switch 语句体内情况标号的值是多少？

B. C 代码中哪些情况有多个标号？

四. （15 分）

一段函数调用的 C 代码如下：

```
#include "stdio.h"

main()
{
    int arg1=718;
    int arg2=415;
    int diff = swap_sub(&arg1, &arg2);
    printf ( "diff=%d\n", diff);
}

int swap_sub(int*xp, int*yp)
{
    int x=*xp;
    int y=*yp;
    int z=0;
    *xp=y;
    *yp=x;
    z=x-y;
    return z;
```

}

要求在下方的栈帧图中（每一格 4 字节）：

地址	内容
0XBFFFF108	OLD EBP
104	
100	
0FC	
0F8	arg2=415
0F4	arg1=718
0F0	
0EC	
0E8	
0E4	
0E0	
0DC	
(EBP) → 0D8	OLD EBP
0D4	
0D0	
0CC	
(ESP) → 0C8	

1. 将“RtnAddr”作为返回地址内容，填入栈帧中的准确位置；
2. 在主函数栈帧中的正确位置，写出传递参数的准确值；
3. 在子函数栈帧中的正确位置，写出子函数执行完毕后各局部变量的准确值；

五. (15 分)

现有包含 main 函数的可执行程序由 GNU OBJDUMP 工具生成的反汇编代码，如下是其中一部分：

```
00000000 <main>:
  0:  8d 4c 24 04      lea    0x4(%esp),%ecx
  4:  83 e4 f0         and    $0xffffffff0,%esp
  7:  ff 71 fc         pushl  0xffffffffc(%ecx)
  a:  55              push   %ebp
  b:  89 e5           mov    %esp,%ebp
  d:  51              push   %ecx
  e:  83 ec 04        sub    $0x4,%esp
11:  e8 fc ff ff ff   call   12 <main+0x12>
             12: R_386_PC32 swap
16:  83 c4 04        add    $0x4,%esp
```

问：

- (1) 此图中每一行冒号前面的部分表示什么内容？此图对应的机器代码可能属于 ELF 的哪个节 (section) ？
- (2) 由此图可知，这里出现了哪种类型的重定位？这种类型的重定位有什么特点？
- (3) 重定位之后，此图哪些部分会发生变化？

六 (20 分)

假定某处理器带有一个数据区容量为128B的数据cache，采用直接映射方式，块大小为32B，主存容量32K。以下C 语言程序段运行在该处理器上，设sizeof(int)=4，编译器将变量i, j, k, sum都分配在通用寄存器中，因此，只要考虑数组元素的访存情况，假定数组a 从第一个主存块开始处存放。请回答下列问题：

- (1) 该cache有多少组？主存地址中的标记位、组索引和块偏移字段分别占几位？
- (2) 当k=15 和k=16 时，执行以下程序的过程中，数据访问不命中率分别是多少？

```
int i, j, k, sum;
int a[64];
for ( i = 0; i < 100; i++){
    for ( j = 0; j < 64; j=j+k ){
        sum+=a[j]; }}

```