

### 附件 3: 试卷格式样张

## 复旦大学计算机科学技术学院 2015 ~2016 学年第二学期期末考试试卷

☐ A 卷      ☐ B 卷

课程名称: 计算机原理      课程代码: COMP130007.01

开课院系: 计算机科学技术学院      考试形式: 开卷 ☒ / 闭卷 / 课程论文 /

姓 名: \_\_\_\_\_ 学 号: \_\_\_\_\_ 专 业: \_\_\_\_\_

题 号	一	二	三	四	五	六	总 分
得 分							

#### 一、选择题 (每小题 2 分, 共 20 分)

- 假定有四个整数用 8 位补码分别表示  $r1 = 0xFE$ ,  $r2 = 0xF2$ ,  $r3 = 0x90$ ,  $r4 = 0xF8$ 。若将运算结果存放在一个 8 位寄存器中, 则下列运算会发生溢出的是 ( )。  
A.  $r1 * r2$   
B.  $r2 * r3$   
C.  $r1 * r4$   
D.  $r2 * r4$
- 浮点数加减运算过程一般包括对阶、尾数运算、规格化、舍入和判溢出步骤。设浮点的阶码和尾数均采用补码表示, 且位数分别为 4 位和 6 位 (均含 1 位符号位)。若有两个数  $X=2^7 * 29/32$ ,  $Y=2^5 * 5/8$ , 则用浮点加法计算  $X+Y$  的结果是 ( )。  
A. 00111 1100010  
B. 00111 0100010  
C. 01000 0010001  
D. 发生溢出
- 某机器字长 16 位, 主存按字节编址, 转移指令采用相对寻址, 由两个字节组成, 第一字节为操作码字段, 第二字节为相对位移量字段。假定取指令时, 每取一个字节 PC 自动加 1。若某转移指令所在主存地址为 2000H, 相对位移量字段的内容为 06H, 则该转移指令成功转移后的目标地址是 ( )。  
A. 2006H  
B. 2007H  
C. 2008H  
D. 2009H

4. 考虑下面的代码，  $M, N$  是被`#define`定义的常量：

```
int mat1[M][N];
int mat2[N][M];

int sum_element( int i, int j ){
    return mat1[i][j] + mat2[j][i];
}
```

编译这段程序，GCC 产生的汇编代码如下：

$i$  位置在 `%ebp + 8`,  $j$  位置在 `%ebp + 12`

```
1 movl 8(%ebp), %ecx
2 movl 12(%ebp), %edx
3 leal 0(%ecx,6), %eax
4 subl %ecx, %eax
5 addl %edx, %eax
6 leal (%edx,%edx,8), %edx
7 addl %ecx, %edx
8 movl mat1(,%eax,4), %eax
9 addl mat2(,%edx,4), %eax
```

根据逆向工程，下列选项正确的是（ ）。

- A.  $M = 7, N = 6$
- B.  $M = 5, N = 9$
- C.  $M = 5, N = 7$
- D.  $M = 9, N = 5$

5. 对于结构声明

```
struct {
    char *a;
    short b;
    double c;
    char d;
    float e;
    char f;
    long long g;
    void *h;
} foo;
```

假设在 Windows 机器上编译它，这里每个  $K$  字节的基本数据结构的偏移量必需是  $K$  的倍数。你可以重新排列这个结构体中的字段，使得结构体所占字节总和变化，则这个结构体（ ）：

- A. 最大占用 40 字节，最少占用 32 字节
- B. 最大占用 48 字节，最少占用 32 字节
- C. 最大占用 56 字节，最少占用 40 字节
- D. 最大占用 48 字节，最少占用 40 字节

6. 由多个源文件组成的 C 程序，经过编辑、预处理、编译，链接等阶段会生成最终的可执行程序。下面哪个阶段可以发现被调用的函数未定义？答：（ ）。  
A. 预处理  
B. 编译  
C. 链接  
D. 执行
7. 下列命中组合情况中，一次访存过程中不可能发生的是（ ）。  
A. TLB 未命中，Cache 未命中，Page 未命中  
B. TLB 未命中，Cache 命中，Page 命中  
C. TLB 命中，Cache 未命中，Page 命中  
D. TLB 命中，Cache 命中，Page 未命中
8. 假定主存地址为 32 位，按字节编址，主存和 Cache 之间采用直接映射方式，主存块大小为 4 个字，每字 32 位，采用回写（Write Back）方式，则能存放 4K 字数据的 Cache 的总容量的位数至少是（ ）。  
A. 146k  
B. 147K  
C. 148K  
D. 158K
9. 下列存储器中，在工作期间需要周期性刷新的是（ ）。  
A. SRAM  
B. DRAM  
C. ROM  
D. FLASH
10. NEMU 作为一个程序，在下列哪一层上运行？答：（ ）。  
A. Micro operating system  
B. GNU/Linux  
C. Computer hardware  
D. Simulated x86 hardware

## 二、（15 分）

1. 考虑下面两种基于 IEEE 浮点数格式的 9 比特位长度的浮点数表示格式
- 格式 A
    - \* 一个符号位
    - \* 有  $k = 5$  个阶码位，阶码偏置量为 15.
    - \* 有  $n = 3$  个小数位
  - 格式 B
    - \* 一个符号位

- \* 有  $k = 4$  个阶码位，阶码偏置量为 7.
- \* 有  $n = 4$  个小数位

下面给出了一些格式 A 表示的位模式，你的任务是把他们转换成最接近的格式 B 表示的值。如果需要舍入，要向正无穷舍入。另外，请给出格式 A 和格式 B 表示的位模式对应的值。要么是整数（例如 17），要么是小数（例如  $17/64$  或  $17/2^6$ ）

Format A		Format B	
Bits	Value	Bits	Value
1 01110 001	-9/16	1 0110 0010	-9/16
0 10110 011			
1 00111 010			
0 00000 111			
1 11100 000			
0 10111 100			

### 三、（22 分）

下面代码是一个函数（不太好）的实现，这个函数从标准输入读入一行，将字符串复制到新分配的存储中，并返回一个指向结果的指针。

```

1 char *getline()
2 {
3     char buf[8];
4     char *result;
5     gets(buf);
6     result = malloc(strlen(buf));
7     strcpy(result, buf);
8     return result;
9 }
```

#### C 语言代码

```

1 80485c0 <getline>:
2 485c0: 55          push %ebp
3 0485c1: 89 e5      mov %esp,%ebp
4 0485c3: 83 ec 28   sub $0x28,%esp
5 0485c6: 89 5d f4   mov %ebx,-0xc(%ebp)
6 0485c9: 89 75 f8   mov %esi,-0x8(%ebp)
7 0485cc: 89 7d fc   mov %edi,-0x4(%ebp)
8 0485cf: 8d 75 ec   lea -0x14(%ebp),%esi
9 0485d2: 89 34 24   mov %esi,(%esp)
10 0485d5: e8 a3 ff ff call 804857d <gets>
```

#### 对 gets 调用的反汇编代码

考虑下面的场景。调用过程 `getline`，返回地址等于 `0x8048643`，寄存器 `%ebp` 等于 `0xbfffc94`，寄存器 `%ebx` 等于 `0x1`，寄存器 `%edi` 等于 `0x2`，而寄存器 `%esi` 等于 `0x3`，输入字符串为 "012345678901234567890123"，程序会因为段错误而中止。运行 GDB，确定错误是在执行 `getline` 的 `ret` 指令时发生。

- A. 填写下图，尽可能多的说明在执行完反汇编代码中第七行指令后栈的相关信息，在右边标出存储在栈中数字的含义，如返回地址，在方框中写出他们的十六进制值，如 `08 04 86 43`。每个方框都代表 4 个字节。并指出 `%ebp` 的位置。（11 分）

<b>08 04 86 43</b>	返回地址

B. 程序应该试图返回到什么地址？（3 分）

C. 当 `getline` 返回时，哪个（些）寄存器的值被破坏了？（8 分）

寄存器	值

四、（20 分）

2. 假设一个 C 语言程序有两个源文件，`main.c` 和 `proc1.c`，它们的内容如下：

```

1  #include <stdio.h>
2  unsigned x = 257;
3  short y, z = 2;
4  void proc1(void);
5  void main()
6  {
7      proc1();
8      printf("x = %u, z = %d\n", x, z);
9      return 0;
10 }
```

main.c 文件

## proc1.c 文件

1) 在上述两个文件中出现的符号哪些是强符号哪些是弱符号。(8分)

3) 修改文件 `proc1.c`, 使得 `main.c` 能够输出正确的结果 (即 `x = 257, z = 2`)。要求修改时不能改变任何变量的数据类型和名字。(4 分)

假定一个计算机系统有一个 TLB 和一个 L1 Data Cache。该系统按字节编址，虚拟地址 14 位，物理地址 12 位，页大小为 64B；TLB 采用 4 路组相联方式，共有 16 个页表项；L1 Data Cache 采用直接映射方式，块大小为 4B，共 16 行。

位	标记位	PPN	有效位	标记位	PPN	有效位	标记位	PPN	有效位	标记位	PPN	有效位
0	03	-	0	09	0D	1	00	-	0	07	02	1
1	03	2D	1	02	-	0	04	-	0	0A	-	0
2	02	-	0	08	-	0	06	-	0	03	-	0
3	07	-	0	03	0D	1	0A	34	1	02	-	0

a) TLB: 四组, 16个条目, 四路组相联

VPN	PPN	有效位	VPN	PPN	有效位
00	28	1	08	13	1
01	-	0	09	17	1
02	33	1	0A	09	1
03	02	1	0B	-	0
04	-	0	0C	-	0
05	16	1	0D	2D	1
06	-	0	0E	11	1
07	-	0	0F	0D	1

b) 页表: 只展示了前 16 个 PTE

索引	标记位	有效位	块 0	块 1	块 2	块 3
0	19	1	99	11	23	11
1	15	0	-	-	-	-
2	1B	1	00	02	04	08
3	36	0	-	-	-	-
4	32	1	43	6D	8F	09
5	0D	1	36	72	F0	1D
6	31	0	-	-	-	-
7	16	1	11	C2	DF	03
8	24	1	3A	00	51	89
9	2D	0	-	-	-	-
A	2D	1	93	15	DA	3B
B	0B	0	-	-	-	-
C	12	0	-	-	-	-
D	16	1	04	96	34	15
E	13	1	83	77	1B	D3
F	14	0	-	-	-	-

c) 高速缓存: 16个组, 4字节的块, 直接映射

针对虚拟地址: 0x03d7 填写下表:

a. 虚拟地址 (2分)

13	12	11	10	9	8	7	6	5	4	3	2	1	0

b. 地址翻译 (6分)

参数	值
虚拟页号	
TLB 索引	
TLB 标记	
TLB 命中? (是/否)	
缺页? (是/否)	
物理页号	

c. 物理地址 (2分)

11	10	9	8	7	6	5	4	3	2	1	0

d. 物理存储器引用 (5分)

参数	值
字节偏移	
缓存索引	
缓存标记	
缓存命中? (是/否)	
返回的缓存字节	

六、（8分）

A. 解释在 PA0 阶段运行 NEMU 会出现 assertion fail 的错误信息的原因，并写出解决方法。（3分）

B. 简述 PA1 阶段中添加和删除监视点的步骤。（5分）