

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: 计算机组成与结构 (2015 春); 试卷编号: A; 考试时间: 120 分钟  
(通信工程 / 信息安全 / 保密管理 / 物联网 / 智能专业)

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

一、(10 分) 请考虑以下两种不同的 6 位浮点格式机器, 采用与 IEEE 浮点标准相同的格式, 即最高位符号位, 接着是阶码, 最后为尾数部分。

(1) 格式 A

- 1 位符号位  $s$ ;
- 3 位阶码  $k$ , 使用基于  $2^{k-1} - 1 = 3$  的偏置值(bias);
- 2 位尾数  $n$ 。

(2) 格式 B

- 1 位符号位  $s$ ;
- 2 位阶码  $k$ , 使用基于  $2^{k-1} - 1 = 1$  的偏置值(bias);
- 3 位尾数  $n$ 。

非规格化数的阶码值  $E = 1 - \text{bias}$ , 规格数的阶码则为  $E = e - \text{bias}$ ; 尾数采取向偶数舍入方式。针对格式 A 和格式 B, 分别填写下表中的二进制编码。

值	格式 A 的编码	格式 B 的编码
0	0 000 00	0 00 000
1		
1/2		
11/8		

二、(30 分) 以下有三段完整或者不完整的 C 程序段，题目给出了它们对应的汇编代码，请利用你掌握的 C 语言和汇编语言知识，采用逆向工程的思维，回答下面的问题。

(1) 请将下面这段缺失的 C 代码填写完整 (8 分)。

C 代码:

```
unsigned mystery1(unsigned n) {  
    if (_____  
        return 1;  
    else  
        return 1 + mystery1(_____  
}
```

汇编代码:

```
mystery1:  
    pushl    %ebp  
    movl     %esp, %ebp  
    subl     $8, %esp  
    cmpl     $0, 8(%ebp)  
    jne      .L2  
    movl     $1, -4(%ebp)  
    jmp      .L3  
.L2:  
    movl     8(%ebp), %eax  
    shrl     %eax  
    movl     %eax, (%esp)  
    call     mystery1  
    addl     $1, %eax  
    movl     %eax, -4(%ebp)  
.L3:  
    movl     -4(%ebp), %eax  
    leave  
    ret
```

(2) 考虑下面数组访问的 C 程序，假设其 H 和 J 值已先行定义:

```
int array1 [H][J];  
int array2 [J][H];  
  
void copy_array (int x, int y) {  
    array2 [y][x] = array1 [x][y];  
}
```

假设上述 C 代码运行于某 x86-64 位的机器上，其汇编代码如下：

```
# on entry:
#   %edi = x
#   %esi = y
#
copy_array:
    movslq %esi, %rsi
    movslq %edi, %rdi
    leaq    (%rsi, %rsi, 8), %rdx
    addq    %rdi, %rdx
    movq    %rdi, %rax
    salq    4, %rax
    subq    %rdi, %rax
    addq    %rsi, %rax
    movl    array1(, %rax, 4), %eax
    movl    %eax, array2(, %rdx, 4)
    ret
```

问：该数组的行值 H 和列值 J 分别为多少？（8 分）

H = \_\_\_\_\_

J = \_\_\_\_\_

(3) 考虑 switch 语句的某种实现，采用 jmpq 指令（64 位机器）来实现跳转表的操作：

```
0x40047b  jmpq    *0x400598(, %rdi, 8)
```

利用 GDB 工具查看内存中的跳转表：

```
0x400598: 0x0000000000400488  0x0000000000400488
0x4005a8: 0x000000000040048b  0x0000000000400493
0x4005b8: 0x000000000040049a  0x0000000000400482
0x4005c8: 0x000000000040049a  0x0000000000400498
```

下面是该 switch 语句的汇编代码：

```
# on entry: %rdx = c and %rsi = b
0x400474: cmp     $0x7, %edi
0x400477: ja      0x40049a
0x400479: mov     %edi, %edi
0x40047b: jmpq    *0x400598(, %rdi, 8)
0x400482: mov     $0x15213, %eax
0x400487: retq
0x400488: sub     $0x5, %edx
```

```

0x40048b: lea    0x0(, %rdx, 4), %eax
0x400492: retq
0x400493: mov    $0x2, %edx
0x400498: and    %edx, %esi
0x40049a: lea    0x4(%rsi), %eax
0x40049d: retq

```

请将以下 C 代码填写完整，C 中的 switch 由上述跳转表实现（14 分）。

```

int main (int a, int b, int c) {
    int result = 4;
    switch (a) {
        case 0:
        case 1:
            _____;
        case: ____
            _____;
            break;
        case: ____
            result = _____;
            break;
        case 3:
            _____;
        case 7:
            _____;
        default:
            _____;
    }
    return result;
}

```

三、(20 分) 某计算机系统主存容量为 256 字节，按字节编址。高速缓存 (Cache) 容量为 32 字节，主存和 Cache 每个数据块大小为 4 字节，试设计一个 2 路组

相联的 Cache，要求：

(1) 指出主存地址中标记 (Tag)，组索引 (Index) 和块偏移 (Offset) 字段的宽度；(8 分)

(2) 假设该 Cache 采用最近最少使用的替换策略。初始是 Cache 内容为空，当 N=10 时，执行下列代码 cache 内实际内容是什么？(int 类型为 4 个字节，v 被加载到从地址 0 开始的连续存储器地址中)

```
Int sumvec( int v[N])
{
    inti, sum=0;
    for ( i=0; i<N; i++ )
        sum += v [i];
    return sum;
}
```

请给出每执行一条指令时，Cache 中内容的变化。(12 分)

四、(30 分) 现有某个简单的虚拟存储系统，其虚拟地址长度为 14bit，物理地址长度为 12bit，页面大小为 64Byte，那么：

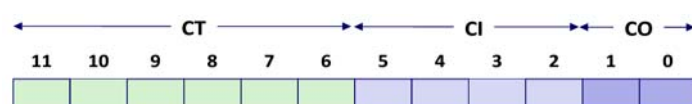
(1) (7 分) 虚拟地址如何组成？物理地址如何组成？页表作用是什么？页表存放在哪里？若有如下页表，其中的 valid 表示什么意思？

VPN	PPN	Valid	VPN	PPN	Valid
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

(2) (16 分) 该系统存在 TLB，TLB 作用是什么？保存在哪里？若 TLB 大小为 16 个条目，采用四路组相联，那么虚拟地址中哪几位表示 TLBT？哪几位表示 TLBI？TLB 内容如下：

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
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

相应系统的高速缓存 Cache 是 16 行，每块大小为 4Byte，采用直接映射，其物理地址形式为：



高速缓存内容如下：

Idx	Tag	Valid	B0	B1	B2	B3	Idx	Tag	Valid	B0	B1	B2	B3
0	19	1	99	11	23	11	8	24	1	3A	00	51	89
1	15	0	—	—	—	—	9	2D	0	—	—	—	—
2	1B	1	00	02	04	08	A	2D	1	93	15	DA	3B
3	36	0	—	—	—	—	B	0B	0	—	—	—	—
4	32	1	43	6D	8F	09	C	12	0	—	—	—	—
5	0D	1	36	72	F0	1D	D	16	1	04	96	34	15
6	31	0	—	—	—	—	E	13	1	83	77	1B	D3
7	16	1	11	C2	DF	03	F	14	0	—	—	—	—

若虚拟地址为 0X03D4，其相应的物理地址是多少？此时 TLB 命中吗？是否发生缺页？若虚拟地址为 0X0020，其相应物理地址是多少？此时 TLB 命中吗？

是否发生缺页？上述，若发生 TLB 不命中，应该怎么办？若发生缺页，应该怎么办？

(3) (7 分) 进程的地址空间是什么样的？代码段和数据段分别在地址空间中的什么地方？





五、（10 分）请写出下列程序的运行结果：

```
#include <stdlib.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>

static inline void read_and_print_one(intfd) {
    char c;
    read(fd, &c, 1);
    printf("%c", c);
    fflush(stdout);
}

int main() {
    int file1 = open("numbers.txt", O_RDONLY);
    int file2;
    int file3 = open("numbers.txt", O_RDONLY);
    file2=dup(file3);
    read_and_print_one(file1);
    read_and_print_one(file2);
    read_and_print_one(file3);
    read_and_print_one(file2);
    read_and_print_one(file3);
    read_and_print_one(file1);
    read_and_print_one(file3);
    return 0;
}
```