武汉大学计算机学院

《计算机系统基础》期末考试试卷

姓名_____ 学号_____

（注：① 开卷考试；② 考试时间为 120 分钟；③ 所有解答必须写在答题纸上。）

注意：
本次试卷全部题目为计算题或论述题，在给出结论的同时，必须给出详细分析和计算过程，否则不能得分。

本试卷上的所有代码，如未特别说明，均在x86-64/Linux机器上编译、链接和执行。

一. （23分）

你负责维护一个C语言编写的程序，其中包含如下代码：

```
1  typedef struct {
2      int first;
3       a_struct a[SIZE];
4      int last;
5  } b_struct;
6
7  void test(long i, b_struct *bp) {
8      int n = bp->first + bp->last;
9      a_struct *ap = &bp->a[i];
10     ap->x[ap->idx] = n;
11 }
```

编译时常量SIZE和结构a_struct的声明你无权访问，不过你有这个程序的目标文件，用objdump反汇编得到如下代码：

```
0000000000000000 <test>:
  0:    55                      push   %rbp
  1:    48 89 e5                mov    %rsp,%rbp
  4:    48 89 7d e8             mov    %rdi,-0x18(%rbp)
  8:    48 89 75 e0             mov    %rsi,-0x20(%rbp)
  c:    48 8b 45 e0             mov    -0x20(%rbp),%rax
 10:    8b 10                   mov     (%rax),%edx
 12:    48 8b 45 e0             mov    -0x20(%rbp),%rax
 16:    8b 80 20 01 00 00       mov    0x120(%rax),%eax
 1c:    01 d0                   add    %edx,%eax
 1e:    89 45 f4                mov    %eax,-0xc(%rbp)
 21:    48 8b 55 e8             mov    -0x18(%rbp),%rdx
```

```
25:   48 89 d0          mov   %rdx,%rax
28:   48 c1 e0 02       shl   $0x2,%rax
2c:   48 01 d0          add   %rdx,%rax
2f:   48 c1 e0 03       shl   $0x3,%rax
33:   48 8b 55 e0       mov   -0x20(%rbp),%rdx
37:   48 01 d0          add   %rdx,%rax
3a:   48 83 c0 08       add   $0x8,%rax
3e:   48 89 45 f8       mov   %rax,-0x8(%rbp)
42:   48 8b 45 f8       mov   -0x8(%rbp),%rax
46:   8b 10             mov    (%rax),%edx
48:   8b 45 f4          mov   -0xc(%rbp),%eax
4b:   48 63 c8          movslq %eax,%rcx
4e:   48 8b 45 f8       mov   -0x8(%rbp),%rax
52:   48 63 d2          movslq %edx,%rdx
55:   48 89 4c d0 08    mov   %rcx,0x8(%rax,%rdx,8)
5a:   90                nop
5b:   5d                pop   %rbp
5c:   c3                retq
```

1) （3分）%rbp、%rsp寄存器的功能是什么？为什么有的编译器编译出的结果里没有使用%rbp？
2) （2分）%rdi和%rsi寄存器里存放的是什么值？
3) （2分）地址0x10开始的那条指令执行完之后，%edx中存放的是什么值？
4) （2分）地址0x16开始的那条指令执行完之后，%eax中存放的是什么值？
5) （2分）地址0x3a开始的那条指令执行完之后，%rax中存放的是什么值？
6) （2分）地址0x46开始的那条指令执行完之后，%edx中存放的是什么值？
7) （2分）地址0x4b开始的那条指令完成了什么功能？
8) （4分）根据反汇编代码推出SIZE的值。（**务必写出推理的过程**）
9) （4分）假设结构a_struct中只有字段idx和x，并且它们都是有符号数值，推出该结构的完整声明。（**务必写出推理的过程**）

二.（17分）

下面的C语言程序实现的功能是计算(float) x，x是一个int类型的变量。

```
1  unsigned float_i2f(int x) {
2    unsigned sign = (x < 0);
3    unsigned ax = (x < 0) ? -x : x;
4    unsigned exp = 127+31;
5    unsigned residue;
6    unsigned frac = 0;
7    if (_____________) {
8      exp = 0;
9      frac = 0;
10   } else {
11     while ((ax & _____________)) == 0) {
12       ax = ax << 1;
13       exp--;
14     }
15     residue = ax & _____________;
16     frac = (ax >> 8) & 0x7FFFFF;
17     if (residue > 0x80 || (residue == 0x80 && (frac & 0x1))) {
18       frac ++;
19       if (frac > 0x7FFFFF) {
20         frac = (frac & 0x7FFFFF) >> 1;
21         exp++;
22       }
23     }
24   }
25   return _____________________________;
26 }
```

1) （8分，每空2分）补充程序中缺失的语句。

   第7行：_____

   第11行：_____

   第15行：_____

   第25行：_____

2) （2分）第17行if语句的判断条件是什么含义？

3) （2分）第19~22行代码的功能是什么？

4) （2分）该函数实现的舍入方法叫做什么方式？这种舍入方式有什么特点？

5) （3分）如果想改成向0舍入，该如何修改代码？

三. （40分）

下图左侧是switch.c的代码，右侧是执行“gcc -S switch.c”得到的代码。

```
/* switch.c */
#include "vector.h"
typedef enum {
   MODE_A, MODE_B, MODE_C, MODE_D,
MODE_E, MODE_F, MODE_G
} mode_t;


void func(int *p1, int *p2, int *p3, int n, mode_t
action) {
   switch (action) {
      case MODE_A:
         ____________①____________
         break;
      case MODE_B:
         ____________②____________
         break;
      case MODE_C:
         ____________③____________
         break;
      case MODE_D:
      case MODE_F:
         p1[0] = (int)(char)p1[0];
      case MODE_E:
         ____________④____________
         break;
      case MODE_G:
         ____________⑤____________
         break;
      default:
```

```
func:
   cmpl  $6, %r8d
         ja      .L2
         subq    $8, %rsp
         movq    %rdx, %r9
         movl    %r8d, %r8d
         leaq    .L4(%rip), %rdx
         movslq  (%rdx,%r8,4), %rax
         addq    %rdx, %rax
         jmp     *%rax

.L4:
         .long   .L3-.L4
         .long   .L5-.L4
         .long   .L6-.L4
         .long   .L7-.L4
         .long   .L8-.L4
         .long   .L7-.L4
         .long   .L9-.L4

.L3:
         movq    %r9, %rdx
         call    addvec@PLT
         jmp     .L1
.L5:
         movq    %r9, %rdx
         call    multvec@PLT
         jmp     .L1
.L6:
         movl    (%rdi), %eax
         cltd
         idivl   (%rsi)
         movl    %edx, (%r9)
         jmp     .L1
.L7:
         movsbl  (%rdi), %eax
         movl    %eax, (%rdi)
.L8:
         movl    (%rdi), %eax
         sarl    $1, %eax
```

_____⑥_____
}
}
	movl %eax, (%r9)
	.L1:
	addq $8, %rsp
	ret
	.L9:
	movl (%rdi), %edx
	movl %edx, %eax
	sall $5, %eax
	addl %edx, %eax
	movl %eax, (%r9)
	jmp .L1
	.L2:
	movl (%rdi), %eax
	movl %eax, (%rdx)
	ret

1) （12分）请根据提供给你的代码，将switch.c文件中func函数的内容填写完整。

① _____ ② _____ ③ _____

④ _____ ⑤_____ ⑥ _____

2) （4分）跳转表放在可重定位目标文件的什么节中？在可执行目标文件被执行时，会加载进内存的什么段中？

3) （2分）如果该switch语句中新增加一个action=100001的处理情况case 100001，你觉得编译器会将跳转表的表项增加到100001项吗？该如何处理呢？

下面的图分别给出了main.c，vector.h，addvect.c和multvec.c的代码。

```
/* main.c */
#define N 3
extern int func(int *, int *, int *, int, int);

int main() {
   int x1[N]={0x1ff,2,1}, x2[N]={4,5,6}, x3[N]={0,0,0};
   func(x1, x2, x3, N, 1);
   return 0;
}
```

```
/* vector.h */
/* prototypes for libvector */
void addvec(int *x, int *y, int *z, int n);
void multvec(int *x, int *y, int *z, int n);
int getcount();
```

```
/* addvec.c */
int addcnt = 0;

void addvec(int *x, int *y, int *z, int n)
{
   int i;
   addcnt++;

   for (i = 0; i < n; i++)
      z[i] = x[i] + y[i];
}
```

```
/* multvec.c */
int multcnt = 0;

void multvec(int *x, int *y, int *z, int n)
{
   int i;
   multcnt++;

   for (i = 0; i < n; i++)
      z[i] = x[i] * y[i];
}
```

4) （8分）用gcc -Og -c main.c得到main.o，根据你的链接知识，填写下表。说明每个标识符是否出现在main.o的符号表（.symtab节）中，如果是的话，进一步说明它的Type（NOTYPE、OBJECT、FUNC、FILE、SECTION）、Bind（LOCAL或GLOBAL），以及它所存放的位置section（.text、.data、.bss或者UND（未定义））。

标识符	.symtab条目？	Type	Bind	Section
N				
func				
main				
x1				

5) （2分）该如何将addvec和multvec打包成动态链接库libvector.so供其他程序使用呢？

6) （4分）我尝试用gcc -Og main.c switch.c编译生成可执行文件，得到如下错误提示：

```
switch.o: In function `func':
switch.c:(.text+0x24): undefined reference to `addvec'
switch.c:(.text+0x2e): undefined reference to `multvec'
collect2: error: ld returned 1 exit status
```

ld是什么？该错误提示是什么意思？该如何解决？

7) （8分）现在我们已经得到了可执行文件zhangsan，根据你的链接知识，填写下表。说明每个标识符是否出现在zhangsan的符号表（.symtab节）中，填表规则同第4小题。

标识符	.symtab条目？	Type	Bind	Section
func				
main				
addvec				
addcnt				

四.（4分）什么是动态链接？相比起静态链接它有什么优点和缺点？

五.（6分）我写了一个.c文件，其中全局变量x统计的是funct函数被调用的次数，形式如下：

```
1  #include “stdio.h”
2
3  int x=0;
4
5  int funct() {
6     ……
7     x++;
8     ……
9     printf(“x = %d\n”, x);
10    ……
11 }
```

我的文件会和其他多个开发者编写的可重定位目标文件一起链接和执行，而我看不到他们编写的程序源码。在执行过程发现，x的值总是错误的，并不能正确统计funct被调用的次数，但是怎么也找不到原因。张三同学很热情的帮我看了一下我的代码，告诉我一定是栈缓冲区溢出造成的，有人有意或无意地修改了我的x的值。

1) （2分）请解释一下什么是栈缓冲区溢出。
2) （4分）我的这个问题有可能是栈缓冲区溢出造成的吗？如果是，请列举对抗该攻击的方法；如果不是，请给出原因，并分析可能是什么原因造成的，有什么方法能解决这个问题？

六.（5分）列举一个使用Amdahl定律解决计算机系统问题的案例。

七.（5分）在计算机系统中，“并行”是很重要的思想。结合我们学习到的计算机系统知识，列举一个计算机系统中现实的“并行”案例，并说明在该案例中，从串行变为并行，可能需要解决哪些问题。