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 闭卷考试；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.（25分）

有如下C语言代码和它对应的汇编代码：

```
1  int f1(unsigned n) {
2      int sum = 1, power = 1;
3      for (unsigned i = 0; i <= n - 1; i++) {
4          power *= 2;
5          sum += power;
6      }
7      return sum;
8  }
```

```
1  f1:
2          subl    $1, %edi
3          movl    $0, %edx
4          movl    $1, %eax
5          movl    $1, %ecx
6  .L2:
7          addl    %eax, %eax
8          addl    %eax, %ecx
9          addl    $1, %edx
10         cmpl    %edx, %edi
11         jnb     .L2
12         movl    %ecx, %eax
13         ret
```

1) 汇编代码的第7行对应C语言代码中的哪条语句？（2分）
2) 汇编代码的第11行语句完成什么功能？能替换成jge吗？为什么？（3分）
3) 我们写了一个main函数，从键盘读入n的值，然后调用*f1(n)*并输出结果。当输入n的值为2时，输出的结果为多少？（2分）
4) 当输入n的值为0后，程序运行了很久都没有结果输出，发生了什么？为什么？（3分）
5) 若将原始代码中n和i的定义都改为int，在输入n=0时，会有结果输出吗？如果没有，请解释为什么。如果有，请说明值是多少。（3分）
6) 若将原始代码中sum和power的类型修改为float，函数的返回值也修改为float，得到函数*f2(n)*。当输入n为23时，*f1(23)*和*f2(23)*的数值是否相等？*f1(23)*是一个整数，它的十六进制表示是多少？*f2(23)*是一个浮点数，它在机器中的表示是什么？（以十六进制表示）？（5分）
7) *f1(24)*和*f2(24)*的值是否相等？*f1(24)*是一个整数，它的十六进制表示是多少？*f2(24)*是一个浮点数，它在机器中的表示是什么？（以十六进制表示）？**（提示：x86-64浮点数采用向偶数舍入的方法）**（5分）
8) 若函数*f2(127)*返回的机器数为7F80 0000H，则其对应的真值是多少？（2分）

二.（20分）

有如下C语言代码和-Og选项编译出的汇编代码：

```
1  long switch_prob(long x, long n) {
2      long result = ______①_____;
3      switch (n) {
4          case 0:
5              result = _____②______;
6              break;
7          case _____③______:
8          case _____④______:
9              result = ______⑤_____;
10             break;
11         case _____⑥______:
12             result = ______⑦_____;
13         case _____⑧______:
14             result = ______⑨_____;
15             break;
16         default:
17             result = _____⑩______;
18       }
19     return result;
20 }
```

```
   # -Og 版本汇编代码
1  switch_prob:
2      cmpq    $4, %rsi
3      je      .L2
4      jg      .L3
5      testq   %rsi, %rsi
6      je      .L4
7      js      .L8
8      subq    $2, %rsi
9      cmpq    $1, %rsi
10     ja      .L11
11     leaq    0(,%rdi,8), %rax
12     ret
13 .L3:
14     cmpq    $101, %rsi
15     je      .L7
16     movl    $1024, %eax
17     ret
19 .L4:
20     leaq    (%rdi,%rsi), %rax
21     ret
22 .L2:
23     xorq    %rsi, %rdi
24 .L7:
25     leaq    1(%rdi), %rax
26     ret
27 .L8:
28     movl    $1024, %eax
29     ret
30 .L11:
31     movl    $1024, %eax
32     ret
```

1) 请填写出C语言代码中空缺的部分。（每个1分，共10分）**（提示：js指令是在SF标志位置1时跳转）**
2) 我们用-O1重新编译了该C语言程序，得到如下汇编代码。-O1版本的第9行，对应C语言版本的哪一行？又对应-Og版本中的哪一行？（4分）
3) -O1版本的第12行cmovb指令完成什么功能？为什么在优化等级较高时会使用cmov指令，它有什么优势？在什么情况下，不适合编译成cmov指令版本？（6分）

```
    # -O1 版本汇编代码
1   switch_prob:
2           cmpq    $4, %rsi
3           je      .L2
4           jg      .L3
5           movq    %rdi, %rax
6           testq   %rsi, %rsi
7           je      .L1
8           subq    $2, %rsi
9           salq    $3, %rdi
10          cmpq    $2, %rsi
11          movl    $1024, %eax
12          cmovb   %rdi, %rax
13          ret
14  .L3:
15          cmpq    $101, %rsi
16          je      .L5
17          movl    $1024, %eax
19          ret
20  .L2:
21          xorq    $4, %rdi
22  .L5:
23          leaq    1(%rdi), %rax
24  .L1:
25          ret
```

三.（12分）

下图左侧的C语言代码转置一个 M X M 矩阵的元素，这里 M 是一个用 #define定义的常数:

```
1  void transpose(long A[M][M]) {
2      long i, j;
3      for (i = 0; i < M; i++)
4          for (j = 0; j < i; j++) {
5              long t = A[i][j];
6              A[i][j] =A[j][i];
7              A[j][i] =t;
8          }
9  }
```

```
1  .L6 :
2      movq (%rdx), %rcx
3      movq (%rax), %rsi
4      movq %rsi, (%rdx)
5      movq %rcx, (%rax)
6      addq $8, %rdx
7      addq $120, %rax
8      cmpq %rdi, %rax
9      jne .L6
```

当用优化等级-O1编译时，得到这个函数的内循环汇编代码如右侧所示，gcc将数组索引转换成了指针代码。

1) 哪个寄存器保存着指向数组元素A[i][j]的指针？（4分）

2) 哪个寄存器保存着指向数组元素A[j][i]的指针？（4分）

3) M的值是多少？（4分）

四．（8分）

小D同学在看书时读到一句话：“通过栈传递参数时，所有的数据大小都向8的倍数对齐”，英文原文为“When passing parameters on the stack, all data sizes are rounded up to be multiples of eight.”他不太明白这句话的意思是：a）每个参数的大小都向8的倍数对齐，还是b）所有的参数大小加在一起，向8的倍数对齐。请你帮他设计一个C语言程序及实验过程，并说明如果观测到什么现象可以推断出哪种理解是正确的。

五.（15分）

为了获得某程序在执行过程中对内存的使用情况，我们想在每次调用malloc()和free()函数时，都输出执行的操作（malloc或free），分配的内存大小及地址或释放的内存地址。我们写了mymalloc.c程序，代码如下所示：

```
1  #include <stdio.h>

2  int count = 0;

3  void *__real_malloc(size_t size);
4  void __real_free(void *ptr);

5  void *__wrap_malloc(size_t size) {
6      count++;
7      void *ptr = __real_malloc(size);
8      printf("malloc(%d) = %p\n", (int)size, ptr);
9      return ptr;
10 }

11 void __wrap_free(void *ptr) {
12     static int count = 0;
13     count++;
14     __real_free(ptr);
15     printf("free(%p)\n", ptr);
16 }
```

1) 小D同学看了代码之后说，这种方法叫做编译时库打桩，需要获得待打桩程序的源码才能工作。他说的对吗？请说明这段代码使用的打桩机制的原理。（3分）

2) 运行gcc -c mymalloc.c命令，会得到mymalloc.o文件。根据你的链接知识，填写下表。说明每个标识符是否出现在mymalloc.o的符号表（.symtab节）中，如果不在，后面的TYPE、BIND和Section以“-”标识；如果在，则说明它的Type（NOTYPE、OBJECT、FUNC、FILE或SECTION）、Bind（LOCAL或GLOBAL），以及它所存放的位置Section（.text、.data、.bss或UND（未定义））。(8分)

标识符	.symtab条目？	Type	Bind	Section
__wrap_malloc				
__real_malloc				
count (第2行)				
count (第12行)				

3) 运行readelf -s mymalloc.o可以看到__wrap_malloc的Size（大小）为81，这是什么意思？count（第2行）的Size为4，这又是什么意思呢？（4分）