



北京邮电大学 2019—2020 学年第 1 学期

《计算机系统基础》期中考试试题

考试 注意 事项	一、学生参加考试须带学生证或学院证明，未带者不准进入考场。								
	二、书本、参考资料、书包等物品一律放到考场指定位置。								
	三、学生不得另行携带、使用稿纸，要遵守《北京邮电大学考场规则》，有考场违纪或作弊行为者，按相应规定严肃处理。								
	四、学生必须将答题内容做在试题答卷上，做在草稿纸上一律无效。								
考试 课程	计算机系统基础			考试时间 2 小时		2019 年 11 月 11 日			
题号	一	二	三	四	五	六	七	八	总分
得分									

一、单项选择题

1. 在 x86-64 机器上，TMin 表示的值为 ()。
- A. -2^{64} B. $-2^{64} + 1$ C. -2^{63} D. $-2^{64} + 1$
2. 执行下列 C 代码后，变量 c 和 d 的值为 ()。
- ```
unsigned a = 0xffffffff;
unsigned b = 2;
int c = a;
int d = b + c;
printf("%d, %d\n", c, d);
```
- A. -1 和 1      B.  $2^{64} - 1$  和 1  
C. -1 和  $2^{64} + 1$       D.  $2^{64} - 1$  和  $2^{64} + 1$
3. 用 gcc 编译 C 语言程序生成在 x86-64 上运行的 64 位程序，给数据类型 short、unsigned、long、int \* 分配的字节数分别是 ( )。
- A. 2、4、8、8      B. 2、4、8、4      C. 2、4、6、8      D. 2、2、8、8
4. 用 gcc 编译下面 C 程序，然后执行，其结果是 ( )。
- ```
#include <stdio.h>

int main(){
    char x = 0x82;
    printf("x = %d\n", x);
    return 0; }
```
- A. x = 130 B. x = 83 C. x = -125 D. x = -126

姓名:

班内序号:

学号:

班级: 20182011

5. 在 Linux 下, 把可执行程序 foo 反汇编生成类似于汇编代码格式的文本文件的命令是 ()。

- A. disas foo foo.s
B. objdump -d foo foo.s
C. disas foo > foo.s
D. objdump -d foo > foo.s

6. 已知 IEEE754 单精度浮点数的值为 0x42E48000, 那它的十进制为 ()。

- A. 114.25
B. 56.125
C. 50.25
D. 25.125

7. 在 linux/x86-64 环境下, 调用函数 func(x, y, z) 的 call 指令执行后存放参数 y 的寄存器是 ()。

- A. %rax
B. %rbx
C. %rsi
D. %rdi

8. 下列关于在 Linux 中 GCC 编译器用法的叙述中, 正确的是 ()。

- A. GCC 第 2 步是预处理, 对 #include、#define 等预处理命令进行处理
B. 执行 gcc foo.c -o bar -Og 命令行, 将编译 foo.c 生成可执行文件 bar
C. 执行 gcc -c -Og foo.c 命令行, 将编译 foo.c 生成汇编文件 foo.s
D. 执行 gcc -Og bar -o foo.c 命令行, 将编译 foo.c 生成可执行文件 bar

9. C 语言有如下声明

```
int array[6] = {10, 20, 30, 40, 50, 60};
```

假设编译器把变量 array 放到 %rcx 寄存器中, 且已知 %rbx 的值为 2, 把 array[2] 的值传送到 %edx 寄存器中的指令是 ()。

- A. leal (%rcx, %rbx, 4), %edx
B. leal (%rbx, %rcx, 4), %edx
C. movl (%rcx, %rbx, 4), %edx
D. movl (%rbx, %rcx, 4), %edx

10. 正确指出下列 3 条指令对错的是 ()。

```
movb $-23, (%ebx), movq (%rbp), (%rbx), cmpq (%rcx, %rdi), %rax
```

- A. 错、对、错
B. 错、对、对
C. 对、对、错
D. 错、错、对

二、若某机器仅为 6 位, 有符号数 (int) 用补码表示, 依据

```
int z;
```

```
int x = -11;
```

```
unsigned y = x;
```

填写下表:

表达式	十进制表示	二进制表示
零	0	000000
z	37	100101
x	-11	110101
y	53	110101
Tmax (补码最大值)	31	011111

三、考虑代码运行在 x86-64 机器上，已知变量 x、y 和 f 定义如下：

```
int x = foo();
```

```
float f;
```

判断下列 C 表达式是否正确。用 O 表示正确，X 表示错误。

编号	C 表达式	
1	$x \ \& \ (x-1) != 0$	X
2	$(x \wedge (x >> 31)) + 1 > 0$	O
3	$x \wedge \sim x < 0$	X
4	$x == (\text{int})(\text{float}) x$	X
5	$f == (\text{float})(\text{double}) f$	O

四、使用 GDB 调试工具调试某程序，表 1 给出了部分内存区间的存储内容，表 2 给出了部分寄存器的值，顺序执行表 3 的每一条指令，回答下列问题。

内存地址	存储内容
0x7ffffffe378:	0x01 0x06 0x40 0x00 0xb8 0x00 0x9a 0x00
0x7ffffffe380:	0x40 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7ffffffe388:	0x00 0x24 0x7b 0xba 0x92 0xcd 0xa6 0x50

表 1

寄存器	值	寄存器	值
%rax	0x0	%rsi	0x7ffffffe380
%rbx	0x0	%rdi	0x64
%rcx	0x0	%rsp	0x7ffffffe378
%rdx	0x7ffffffe488	%rip	0x4005b6

表 2

指令	回答下列问题
lea (%rdi,%rdi,8),%rdx	(1) %rdx = 400
lea (%rdx,%rdx,1),%rax	(2) %rax = 1800
add \$0x10,%rax	(3) ZF = 0
mov (%rsi),%rcx	(4) %rcx = 7 0x90
lea 0x7(%rcx),%rdx	(5) %rdx = 0x7ffffffe37f 0x47
test %rcx,%rcx	(6) SF = 1
cmovns %rcx,%rdx	(7) %rdx = 0x47
sar \$0x3,%rdx	(8) %rdx = 0x08
mov %rdx,(%rsi)	(9) (%rsi) = 0x08

表 3

3 0100 0111

0000/000

五、C 语言函数 fabc 的部分代码如左下表所示，用 GCC 编译之生成汇编代码如右下表所示，分析汇编代码，完成 (1) ~ (5) 填空。

C 代码

汇编代码

```
int fabc (int *array, int size) {
    int i, lucky;
    for (i = 0; i < size; i++)
    {
        lucky = array[i];
        if (i == lucky)
        {
            (1) lucky = lucky - i;
            return (2) lucky;
        } else
        {
            (3) array[i+1];
            (4) i = i + 2;
        }
    }
    return (5) -1;
}
```

```
fabc:
    movl    $0, %eax
    jmp     .L2
.L5:      movslq  %eax, %rdx
    leaq    (%rdi, %rdx, 4), %r8
    movl    (%r8), %ecx
    cmpl    %ecx, %eax
    jne     .L3
    subl    %eax, %ecx
    movl    %ecx, (%r8)
    ret
.L3:      leaq    4(%rdi, %rdx,
4), %r9
    movl    (%r9), %edx
    addl    $211, %edx
    movl    %edx, (%r8)
    movl    %ecx, (%r9)
    addl    $1, %eax
.L2:      cmpl    %esi, %eax
    jl      .L5
    movl    $-1, %eax
```