

北京邮电大学 2024-2025 学年第一学期

《计算机系统基础》期中考试试题

重制: Hexropt

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 int, 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. 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`
5. 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
6. 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		100101
x	-11	
Y		
Tmax (补码最大值)		

三、考虑代码运行在 x86-64 机器上，已知变量 x 和 f 定义如下：

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

```
float f;
```

判断下列 C 表示式是否正确。用 O 表示正确，X 表示错误。

编号	C 表达式	回答问题
1	$x \& (x-1) != 0$	
2	$(x^{(x \gg 31)}) + 1 > 0$	
3	$x^{\sim} x < 0$	
4	$x == (\text{int})(\text{float})x$	
5	$f == (\text{float})(\text{double})f$	
6	$f == -(-f);$	

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
0x7fffffff378	0x01 0x06 0x40 0x00 0xb8 0x00 0x9a 0x00
0x7fffffff380	0x40 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x7fffffff388	0x00 0x24 0x7b 0xba 0x92 0xcd 0xa6 0x50

表 1

寄存器	值	寄存器	值
%rax	0x0	%rsi	0x7fffffff380
%rbx	0x0	%rdi	0x64
%rcx	0x0	%rsp	0x7fffffff378
%rdx	0x7fffffff488	%rip	0x4005b6

表 2

指令	回答问题
lea (%rdi,%rdi,8),%rdx	(1)%rdx=
lea (%rdx,%rdx,1),%rax	(2)%rax=
add \$0x10,%rax	(3)ZF 标志=
Mov (%rsi),%rcx	(4)%rcx=
lea 0x7(%rcx),%rdx	(5)%rdx=
test %rcx,%rcx	(6)SF=
cmovns %rcx,%rdx	(7)%rdx=
sar \$0x3,%rdx	(8)%rdx=
mov %rdx,(%rsi)	(9)(%rsi)=

表 3

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 语言	汇编代码
<pre> int fabc (int *array,int size){ int i,lucky; for(i=0;i<size;i++){ lucky=array[i]; if(i==lucky) { (1)____; return(2)____; }else { (3)____; (4)____; } } return(5)____; } </pre>	<pre> 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 ret </pre>