---

1. CPU 中的译码器主要用于 B
A 地址译码;
B 指令译码;
C 选择多路数据至 ALU;
D 数据译码
ID: 9504

---

2. 假设将一个 4 位数值（用十六进制数字 0-F 表示）截断到一个 3 位数值，则无符号数原始值为 0，那它的截断值为 D
A 1
B 2
C 3
D 0
ID: 9524

---

3. 假设寄存器为 8 位，用补码形式存储机器数，包括一位符号位，那么十进制数-25 在寄存器表示为 D
A 67H
B 99H
C E6H
D E7H
ID: 9525

---

4. 假设寄存器%eax 的值为 x,%ecx 的值为 y 则 leal 9(%eax,&ecx,2),%edx 操作后寄存器%edx 的值为 B
A 9(x+2y)
B 9+x+2y
C 9+2x+y
D 9+2(x+y)
ID: 9526

---

5. 以下叙述中( )是错误的。 B
A 取指令操作是控制器固有的功能，不需要在操作码控制下完成
B 所有指令的取指令操作都是相同的
C 在指令长度相同的情况下，所有指令的取指操作都是相同的
D 一条指令包含取指、分析、执行三个阶段
ID: 9527

---

6. 浮点数的表示范围和精度取决于( )。 C

A 阶码的位数和尾数的机器数形式

B 阶码的机器数形式和尾数的位数

C 阶码的位数和尾数的位数

D 阶码的机器数形式和尾数的机器数形式

ID: 9528

---

7. 基址寻址方式中，操作数的有效地址是( )。 A

A 基址寄存器内容加上形式地址（位移量）

B 程序计数器内容加上形式地址

C 变址寄存器内容加上形式地址

D 寄存器内容加上形式地址

ID: 9529

---

8. 直接寻址的无条件转移指令功能是将指令中的地址码送入( )。 A

A PC

B 地址寄存器

C 累加器

D ALU

ID: 9530

---

9. 在浮点数编码表示中，( )在机器数中不出现，是隐含的。 A

A 基数

B 尾数

C 符号

D 阶码

ID: 9531

---

10. 下列叙述中概念正确地是( ) D

A 定点补码运算时,其符号位不参加运算

B 浮点运算中,尾数部分只进行乘法和除法运算

C 浮点数的正负由阶码的正负符号决定

D 在定点小数一位除法中为了避免溢出被除数的绝对值一定要小于除数的绝对值

ID: 9532

若浮点数的阶码和尾数都用补码表示，则判断运算结果是否为规格化数的方法是( ) C

11.
A 阶符与数符相同为规格化数
B 阶符与数符相异为规格化数
C 数符与尾数小数点后第一位数字相异为规格化数
D 数符与尾数小数点后第一位数字相同为规格化数
ID: 9533

指令系统中采用不同寻址方式的目的主要是( ) C

12.
A 实现存储程序和程序控制
B 可以直接访问外存
C 缩短指令长度，扩大寻址空间，提高编程灵活性
D 提供扩展操作码的可能并降低指令译码难度
ID: 9534

寄存器间接寻址方式中，操作数在( )。 B

13.
A 通用寄存器
B 主存单元
C 程序计数器
D 堆栈
ID: 9535

当调用 malloc 这样的 C 标准库函数时，( )可以在运行时动态的扩展和收缩。 B A

14.
A 栈
B 堆
C 共享库
D 内核虚拟存储器
ID: 9536

假设寄存器%eax 的值为 x,%ecx 的值为 y，那么汇编代码指令 leal(%eax，%ecx，5），%edx 存储在寄存器%edx 中的值为( )。 D

15.
A 5x
B 5y
C 5x+y
D 5y+x
ID: 9537

16. ---

假设初始值：%dh=CD，%eax=98765432,则执行 movzbl　%dh ,%eax 这样一条指令后，%eax 的值为( ) D

A %eax= 987654CD
B %eax= CD765432
C %eax= FFFFFFCD
D %eax= 000000CD
ID: 9538

---

17. 假设初始值：%dh=CD，%eax=98765432,则执行 movsbl　%dh ,%eax 这样一条指令后，%eax 的值为( ) C

A %eax= 987654CD
B %eax= CD765432
C %eax= FFFFFFCD
D %eax= 000000CD
ID: 9539

---

18. 假设一个 4 位数值（用十六进制数字 0~F 表示）截断到一个 3 位数（用十六进制 0~7 表示），[1011]截断后的补码值是 B

A -3
B 3
C 5
D -5
ID: 9523

---

19. 假设在 C 程序中有 int *a , int n ,如果值 a 在寄存器%ecx 中，n 在%edx 中，下面哪个指令计算的是 a[n]？ C

A ret (%ecx,%edx,4)
B leal (%ecx,%edx,4),%eax　ret
C mov (%ecx,%edx,4),%eax ret
D mov (%ecx,%edx,1),%eax ret
ID: 9522

---

20. 将二进制 0.001101 化为十进制数为 C

A 0.25
B 0.1875
C 0.203125
D 0.1992157
ID: 9521

21. ---

下列说法错误的是 D

A 任何二进制整数都可用十进制表示;
B 任何二进制小数都可用十进制表示;
C 任何十进制整数都可用二进制表示;
D 任何十进制小数都可用二进制表示。
ID: 9505

---

22. 下列指令不会改变条件码的值的是 D

A testl %eax,%eax
B addl %eax,%eax
C cmpl %esi,%dsi
D jge .L2
ID: 9506

---

23. 下面哪项是错误的 C

A 由于表示的精度有限，浮点运算不具备结合性，一般选择最小的先运算
B 有符号数遇见无符号数会默认强制转换为无符号数
C 补码编码是表示无符号整数的最常见的方式
D 浮点数编码是表示实数的科学计数法的以二为基数的版本
ID: 9507

---

24. 在补码的加法中发生了负溢出的是 D

A x+y = x+y+2^(w-1)
B x+y = x+y-2^(w-1)
C x+y = x+y-2^w
D x+y = x+y+2^w
ID: 9508

---

25. 下列汇编指令中正确的是 A

A movl $0x4050,%eax
B movl(%eax),4(%esp)
C movl %eax,$0x123
D movb $0xF,%ebx
ID: 9509

---

26. 在下列指令中，会影响条件码中的 CF 位的是 C

A JAE NEXT
B INC BX

C SHL AX,1
D JMP NEXT
ID: 9510

根据向偶数舍入的规则，说明二进制小数 10.010 ， 10.110 ， 11.011 分别舍入到最接近的二分之一（二进制小数点右边一位）的结果是 C

27.

A 10.1，10.1，11.1
B 10.1，11.0，11.0
C 10.0，11.0，11.1
D 10.0，10.1，11.1
ID: 9511

某计算机内存空间按字节编址，若某区域的起始地址为：4A000H,终止地址为 4DFFFH，则该段内存区域的容量是 A

28.

A 16KB
B 256KB
C 1MB
D 2MB
ID: 9512

某计算机存储器按字节编址，采用小端方式存放数据。假定编译器规定 int 型和 short 型长度分别为 32 位和 16 位并且数据按边界对齐存储某 C 语言程序段如下：

```
struct{
    int a;
    char b;
    short c;
}
record;
record.a=273;
```

29.

若 record 变量的首地址为 0XC008，则低地址 0XC008 中内容及 record.c 的地址是 B

A 0X00、0XC00D
B 0X11、0XC00E
C 0X11、0XC00D
D 0X00、0XC00E
ID: 9513

30.

两个补码数相加，可能产生溢出的情况是 A

A 符号位相同

B 符号位不同

C 两个正数相加结果为正

D 数值位产生向符号位的进位，符号位也向更高位产生进位

ID: 9514

考虑下列代码:

unsigned u = 0xFFFFFFFF;

int tu = (int)u;

31. 执行完后 tu 的值是多少? C

A 0xFFFFFFFF

B 1

C -1

D 4294967295

ID: 9515

---

32. 32 位系统中，将一个双字值压入栈中，首先需要将栈指针 A

A 减 4

B 减 2

C 加 4

D 加 2

ID: 9516

---

33. 将一个 4 位数值-5 截断到 3 位数的结果为 C

A -5

B 5

C 3

D -3

ID: 9517

---

34. 将十进制数 167 用十六进制表示的结果是 B

A 0XB7

B 0XA7

C 0XB6

D 0XA6

ID: 9518

---

35. 将二进制 1101011011011111100110 转换成十六进制[B]
A 45A7E6
B 35B7E6
C 35A776
D 56A8E7
ID: 9519

---

36. 将二进制 1011011110011100 转换成十六进制[B]
A B89C
B B79C
C 7B99
D 78BC
ID: 9520

---

37. 假设初始值：%dh=CD，%eax=98765432,则执行 movb %dh ,%al 这样一条指令后，%eax 的值为( )[A]
A %eax= 987654CD
B %eax= CD765432
C %eax= FFFFFFCD
D %eax= 000000CD
ID: 9540

---

38. 假设初始时%dh = CD, %eax = 98765432,
movb %dh, %al movsbl %dh, %eax movzbl %dh, %eax 这三条指令的%eax 的值分别是( )。[C]
A A.987654CD FFFFFFCD FFFFFFCD
B B.98765400 FFFFFFCD FFFFFFCD
C C.987654CD FFFFFFCD 000000CD
D D.987654CD 000000CD FFFFFFCD
ID: 9541

---

39. 假设 x 和 y 的字节值分别为 0x66 和 0x39，计算表达式 x&&~y 的字节值为( )[A]
A 0x01
B 0x10
C 0x11
D 0x21
ID: 9542

---

40. 二进制 11001110 执行算术右移（SAR）一位得到 [A]
A 11100111
B 1100111
C 11100110
D 1100110
ID: 9562

---

41. 对整数运算 z=x+y,设置条件码 OF 的表达式为 [D]
A （unsigned） z < （unsigned） x
B z == 0
C z < 0
D (x < 0 == y < 0) && (z < 0 != x < 0)
ID: 9563

---

42. 对整数运算 z=x+y,设置条件码 CF 的表达式为 [A]
A （unsigned） z < （unsigned） x
B z == 0
C z < 0
D (x < 0 == y < 0) && (z < 0 != x < 0)
ID: 9564

---

43. 对长度为 4 位的整数数据，-5 对应的补码编码为 [A]
A 1011
B 1101
C 101
D 1010
ID: 9565

---

44. 对于指令 MOVL $0x23,(%eax) ,下列说法正确的是 [D]
A 将立即数 23 传送至寄存器 EAX
B 该指令不能执行，有语法错误
C 将立即数 23 传送至 EAX 寄存器中的保存的内存地址
D 将立即数 35 传送至 EAX 寄存器中的保存的内存地址
ID: 9566

---

45. 对于一个无符号数字 X，截断它到 K 位的结果就相当于计算 [A]
A x mod 2 的 K 次方

B x mod 2 的（K-1）次方
C x mod 4 的 K 次方
D x mod 2 的（K-2）次方
ID: 9567

---

46. 对于我们熟知的 ZF，从条件码的角度看，它指的是 B
A 进位标志
B 零标志
C 符号标志
D 溢出标志
ID: 9568

---

47. 对于我们熟知的 OF，从条件码的角度看，它指的是 D
A 进位标志
B 零标志
C 符号标志
D 溢出标志
ID: 9569

---

48. 对于我们熟知的 CF，从条件码的角度看，它指的是 A
A 进位标志
B 零标志
C 符号标志
D 溢出标志
ID: 9570

---

49. 对于某台计算机，以下说法正确的是 C
A int 型值与 short 型值长度总是不一样
B int 型值与 float 型值长度总是一样
C 指向 int 型的指针与指向 char 型的指针长度总是一样
D double 型值与 float 型值总是不一样
ID: 9571

---

50. 对于逻辑运算中，下列表达式的结果为 0x00 的是 B
A !0x00
B !0x41
C 0x69&&0x55
D 0x69||0x55
ID: 9572

---

51. 对于汇编指令 popl，一般与之搭配的指令是 C

A movl
B call
C pushl
D add
ID: 9573

---

52. 对于单精度浮点数 float，下列说法错误的是 B

A frac 为 0 位到第 22 位，即 2^23 = 8388608，一共七位，这意味着最多能有 7 位有效数字，但绝对能保证的为 6 位，所以 float 的精度为 6~7 位有效数字；
B exp 为第 23 到第 30 位，当全取 1 时为无穷大；
C float 的指数范围为-126~+127；
D 负指数决定了浮点数所能表达的绝对值最小的非零数，正指数决定了浮点数所能表达的绝对值最大的数，也即决定了浮点数的取值范围；

ID: 9574

---

53. 对于 32 位机器，char*的字节数为 C

A 1
B 2
C 4
D 8
ID: 9575

---

54. 对跳转指令中 jae .L1 的描述正确的是 D

A 小于或相等时跳转到.L1(有符号>=)
B 小于或相等时跳转到.L1(无符号>=)
C 超过或相等时跳转到.L1(有符号>=)
D 超过或相等时跳转到.L1(无符号>=)
ID: 9576

---

55. 对于 mov 指令，源操作数和目的操作数的组合错误的是 C

A 寄存器到内存数据传送
B 寄存器到寄存器数据传送
C 内存到内存数据传送
D 内存到寄存器数据传送

ID: 9577

在 64 位机器中，将一个 double 型数一次性完整的传入某个寄存器中，应该使用的指令是 D

56.

A movl
B movb
C movw
D movq
ID: 9561

假设寄存器%rax 的值为 x, %rcx 的值为 y；有表达式 leaq 9(%rax,%rcx,2), %rdx；请选择%rdx 中的值 B

57.

A 9*(x+y+2)
B 9+x+2*y
C 9*(x+2*y)
D 9+x+y+2
ID: 9560

寄存器%rax 中的值是 16，请问以下哪个操作后寄存器中%rdx 中值不等于 16 A

58.

A movq %rax (%rdx)
B movq %rax %rdx
C movq $16 %rdx
D movq $0x10 %rdx
ID: 9559

假设 x 和 y 的字节值分别为 0x66 和 0x39，计算表达式!x||!y 的字节值为( )。 B

59.

A 0x11
B 0x00
C 0x01
D 0x10
ID: 9543

假设 AL=5H，要使得 AL=0FAH，应执行的命令是( )。 A

60.

A NOT %AL
B AND $0x0FH，%AL
C XOR 0xF0H,%AL
D OR $0x0FH,AL

ID: 9544

---

61. 假设%edx 的值为 a，%eax 的值为 b，执行下面三条指令
(1)cmpl %eax， %edx
(2)setl %al
(3)movzbl %al,%eax 这三条指令中(3)代表( )意思。 C
A 比较 a 和 b 的大小
B 清零%eax
C 清零%eax 的三个高位字节
D 置%eax 的低字节为 0 或 1
ID: 9545

---

62. 假设%eax 的值为 0x100，%edx 值为 0x3；
地址 0x100,0x104,0x108,0x10C 的值分别是 0xFF,0xAB,0x13,0x11;
则执行指令：imull $16,(%eax,%edx,4) 后，被更新的储存器和更新值分别是多少？ D
A 0x100，0xFF0
B 0x104，0xAB0
C 0x108，0x130
D 0x10C，0x110
ID: 9546

---

63. 计算机系统中采用补码运算的目的是为了( ) C
A 与手工运算方式保持一致
B 提高运算速度
C 简化计算机的设计
D 提高运算的精度
ID: 9547

---

64. 计算机能直接识别和执行的语言是( ) C
A 高级语言
B 汇编语言
C 机器语言
D 自然语言
ID: 9548

---

65. 计算机操作的最小时间单元为( ) A
A 时钟周期
B 指令周期

C CPU 周期
D 中断周期
ID: 9549

---

66. 计算 Imm(Eb ,Es ,s)这种寻址模式所表示的有效地址为( ) A
A Imm + R[Eb]+R[Es] *s
B Imm + R[Eb]+R[Es]
C Imm + R[Eb]
D Imm +R[Es]
ID: 9550

---

67. 汇编指令 LOOPE/LOOPZ 是指( ) C
A CX 不为零且标志 ZF=0 时循环
B CX 为零且标志 ZF=0 时循环
C CX 不为零且标志 ZF=1 时循环
D CX 为零且标志 ZF=1 时循环
ID: 9551

---

68. 设 32 位机器中存在联合体
Union a{ int p[3]; Union a *next; double b}; Union a 存储空间占用的字节数 B
A 40
B 24
C 12
D 8
ID: 9552

---

69. 64 位机器中声明 int *p[8],其占用的存储空间是多少字节 B
A 32
B 64
C 8
D 40
ID: 9553

---

70. 在 32 位机器中声明 char (*p)[8],其占用的存储空间是多少字节 D
A 32
B 16
C 8
D 4

ID: 9554

---

71. 在 IA32 过程调用中，一组统一的寄存器被用来存储调用者和被调用者的数据，通常来说，调用者保存寄存器为下面哪一组 A

A %eax %edx %ecx
B %eax %ebx %ecx
C %ebx %esi %edi
D %ebx %ecx %edx
ID: 9555

---

72. 在 C 语言过程调用机制中的压栈过程中，下面哪项最先入栈 B

A 局部变量
B 返回地址
C 被保存的寄存器
D 参数构造区
ID: 9556

73. 跳转指令 ja 的跳转条件为下面哪个条件码的组合是 B

A ~(SF^OF)
B ~CF&~ZF
C CF|ZF
D SF^OF)&~ZF
ID: 9557

---

74. 跳转指令 jge 的跳转条件为下面哪个条件码的组合 B

A SF^OF
B ~(SF^OF)
C ~(SF^OF)&~ZF
D (SF^OF)|ZF
ID: 9558

---

75. 浮点数中，最大的规格化正数为 D

A 阶码为 00…00，尾数为 11…11
B 阶码为 01…11，尾数为 00…00
C 阶码为 11…11，尾数为 11…11
D 阶码为 11…10，尾数为 11…11
ID: 9578

76. ---

cmova %eax,%ebx 在什么情况下会将%eax 的值传送给%ebx?

B

A CF=1 and ZF=0

B CF=0 and ZF=0

C CF=0 and SF=0

D CF?ZF=1

ID: 9427

两补码相加，采用 1 位符号位，当( )时，表示结果溢出 D

A 符号位有进位

77. B 符号位进位和最高数位进位异或结果为 O

C 符号位为 1

D 符号位进位和最高数位进位异或结果为 1

ID: 9447

下面(??)是对处理器、主存和 I/O 设备的抽象表示 A

A 进程

B 虚拟存储器

C 文件

D 虚拟机

78. ID: 9448

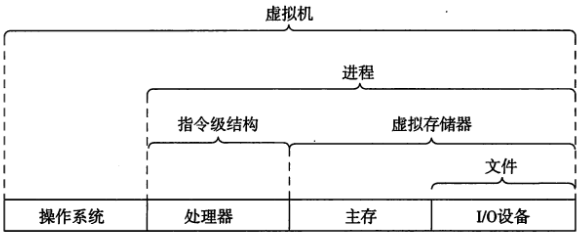

图 1-18 计算机系统提供的一些抽象

下列（）条件码可以用来检查无符号操作的溢出 A

A CF

79. B ZF

C SF

D OF

ID: 9449

在下列指令中，指令的执行会影响条件码中的 CF 位 D

A JMP NEXT

80. B JE NEXT

C INC BX

D SHL AX,1

ID: 9450

---

81. 若 x=103，y=-25，则下列表达式采用 8 位定点补码运算实现时，会发生溢出的是 C

A x+y

B -x+y

C x-y

D -x-y

ID: 9451

---

82. 以下不是常用条件码的有： D

A CF

B ZF

C SF

D GF

ID: 9452

---

83. 机器代码提供两种基本的低级机制来实现有条件的行为：测试数据值然后根据测试的结果来改变控制流或数据流 A

A 正确

B 错误

ID: 9453

---

84. 加载有效地址指令 leal 实际上是 movl 指令的变形。它的指令形式是从存储器读数据到寄存器，过程中引用的存储器。 B

A 正确

B 错误

ID: 9454

---

85. 下列不属于操作数类型的是（） C

A 立即数

B 存储器

C 计数器

D 寄存器

ID: 9455

---

86. 一个 IA32 中央处理单元包含一组 8 个存储 32 位值的寄存器，这些寄存器用来存储（） A

A 整数数据和指针

B 浮点数据和地址
C 整数数据和标志符
D 浮点数据和指针
ID: 9456

---

87. 下列对于 Intel 和 ATT 格式之间所存在的不同的描述正确的是（）
C
A ATT 代码省略了指令大小的后缀
B 仅在带有多个操作数的指令情况下，列出的操作数顺序相同
C Intel 代码不同的方式来描述存储器中的位置
D Intel 代码省略了寄存器名字前面的‘&’符号
ID: 9457

---

88. 在大多数机器上，整数除法要比整数乘法更快
B
A 正确
B 错误
ID: 9458

---

89. c 语言中，当声明两个整数 int i,j 时，i 和 j 在内存中分配的地址肯定相邻，这种说法
B
A 正确
B 错误
ID: 9459

---

90. 单精度和双精度浮点数分别在 32 位机器中使用（）字节。
B
A 2,4
B 4，8
C 8，16
D 12,32
ID: 9460

---

91. 计算机使用（）来指明整数和指针数据的标称大小。
C
A 字
B 字节
C 字长
D 比特
ID: 9461

92. ---

char *p 指针 p 占多少个字节 D

A 1

B 2

C 3

D 4

ID: 9462

---

93. 对于一个无符号数字 x,截断它到 k 位的结果是 B

A x mod k

B x mod 2k

C C.x /k

ID: 9446

位于存储器层次结构中的最顶部的是 A

A 寄存器

B 主存

C 磁盘

D 高速缓存

ID: 9445

94.

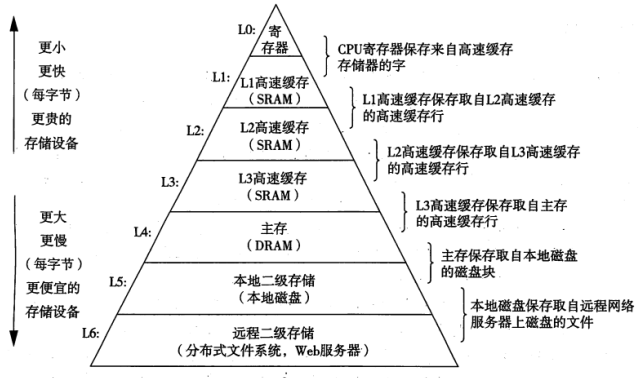

图 1-9　一个存储器层次结构的示例

处理器在指令的要求下将一个字节从主存复制到寄存器的操作是 B

95. A 存储

B 加载

C 操作

D 跳转

ID: 9444

---

96. CMP 指令的运行对条件码和目标寄存器产生的作用是 C

A 设置条件码和更新目标寄存器

B 不设置条件码和更新目标寄存器
C 设置条件码和不更新目标寄存器
D 不设置条件码和不更新目标寄存器
ID: 9428

---

97. CPU 的组成中不包含[B]
A 运算器
B 存储器
C 控制器
D 寄存器
ID: 9429

---

98. CPU 要访问的某一存储单元的实际地址称[C]
A 段地址
B 逻辑地址
C 物理地址
D 偏移地址
ID: 9430

---

99. CPU 有一个程序计数器 PC 它用于存储[B]
A 保存当前 CPU 访问的内存地址
B 保存提取下一条指令的地址
C 暂时存放 ALU 运算信息
D 保存当前正在执行的一条指令
ID: 9431

---

100. C 语言程序在编译运行的过程中会产生一系列中间文件，下列与 hello 程序的相关文件中，是二进制文件不能直接文本编辑器打开查看的是[D]
A hello.s
B hello.i
C hello.c
D hello.o
ID: 9432

---

101. double *D[5]的元素大小和总大小分别为 A

A 8 40

B 8 5

C 4 20

D 4 5

ID: 9433

---

102. gcc 编译程序时，需要在执行文件中产生调试文件的 gcc 附加参数是 B

A (-o)

B (-g)

C (-S)

D (-E)

ID: 9434

---

103. IA32 指令集中 MOV 类指令不包含以下哪一条？ C

A movb

B movw

C movx(外部数据传送)

D movl

ID: 9435

---

104. 一条计算机指令中通常包含 B

A 字符和数据

B 操作码和操作数

C 运算符和数据

D 被运算数和结果

ID: 9436

---

105. IEEE 浮点数表示数时会划分为三个字段，其中表示阶码字段的是 B

A s

B exp

C frac

D f

ID: 9437

106. ---

IEEE 浮点数用 V = （-1）^s * M * 2^E 的形式来表示一个数，其中 E 表示的是[D]

A 符号

B 正整数数阶码

C 尾数

D 可正可负整数阶码

ID: 9438

---

107. int x = 2; int y = 3; x= (x > y) ? x>>1:y>>1,执行后 x =[A]

A 1

B 2

C 6

D 3

ID: 9439

---

108. leal 6(%eax),%edx 操作的结果是[B]

A 6x

B 6+x

C 6-x

D x-6

ID: 9440

---

109. leal 7(%edx,%ebx,5),%eax 假设 %ebx 的值为 y， %edx 的值为 x， 则%eax 的值为[A]

A 5*y+x+7

B 5*x+y+7

C 7*y+x+5

D 7*x+y+5

ID: 9441

---

110. 下面关于 intel 汇编代码格式说法错误的是[A]

A 带有指示大小的后缀

B 省略了寄存器名字前缀 %

C 可以用[ebp+8]描述存储器中的位置

D 和 ATT 格式列出的操作数顺序相反

ID: 9442

---

111. 8086CPU 在基址加变址的寻址方式中,变址寄存器可以为[D]

A BX 或 CX

B CX 或 SI
C DX 或 SI
D SI 或 DI （源变址/目的变址寄存器）
ID: 9443

---

112. call 指令的效果是将返回地址入栈,并跳转到被调用过程的起始处.返回地址是紧跟在程序中 call 后面的那条指令的地址( ) A
A 对
B 错
ID: 9463

---

113. a=01101001,b=01010101,则 a^b=( )。 B
A 1000100
B 111100
C 1011100
D 1011100
ID: 9464

---

114. a=01101001,b=01010101,则 a&b=( ) A
A [01000001]
B [10110101]
C [01011111]
D [11011010]
ID: 9465

---

115. 下面对 show_bytes 的调用将输出什么结果？( )

```
#include <stdio.h>
typedef unsigned char *byte_pointer;
void show_bytes(byte_pointer start,int len){
    int i;
    for(i=0;i<len;i++){
        printf(" %.2x",start[i]);
}
printf("\n");
}

const char *s="abcdef";
show_bytes( (byte_pointer)s, strlen(s) );
```

提示：‘a’-‘z’的 ASCII 码为 0x61-0x7A

```
const char *s="abcdef";
```

show_bytes((byte_pointer)s, strlen(s));提示：‘a’-‘z’的 ASCII 码为 0x61-0x7A C

A a b c d e f

B 66 65 64 63 62 61

C 61 62 63 64 65 66

D f e d c b a

ID: 9485

---

116. 将一个 c 文件翻译成一个可执行目标文件，这个翻译的过程可分为四个阶段完成，下列四个阶段排序正确的是？（ ） A

A 预处理阶段，编译阶段，汇编阶段，链接阶段

B 预处理阶段，汇编阶段，编译阶段，链接阶段

C 预处理阶段，汇编阶段，链接阶段，编译阶段

D 预处理阶段，编译阶段，链接阶段，汇编阶段

ID: 9486

---

117. 汇编语言源程序中，每个语句由四项组成，如语句要完成一定功能，那么该语句中不可省略的项是( ) B

A 名字项

B 操作项

C 操作数项

D 注释项

ID: 9487

---

118. 汇编语言语句格式中，对名字项的规定，请指出错误的解释( ) B

A 名字的第一个字符可以是大小写英文字母及？、@、__等

B 名字的第一个字符可以是大小写英文字母、数字、？、@、__等

C 名字的有效长度≤31 个字符

D 名字从第二个字符起可以出现数字，但不允许出现#等字符

ID: 9488

---

119. 汇编代码后缀 I 表示( ) C

A 字节

B 字

C 双字

D 位

ID: 9491

120. ---

185.过程 P 调用过程 Q 时，假设返回地址在内存中的地址为 X，%ebp 值保存在内存地址 Y，则以下说法正确的是（32 位环境下）：

B

A X=Y-4

B X=Y+4

C X=Y+8

D X=Y-8

ID: 9492

过程 P 调用过程 Q 时，Q 有两个参数，假设返回地址在内存中的地址为 X，传递给 Q 的第二个参数保存在内存地址 Y，则以下说法正确的是（32 位环境下）

B

121.

A X=Y-4

B X=Y-8

C X=Y+4

D X=Y+8

ID: 9493

过程 P 调用过程 Q 后，记过程 P 栈帧的%ebp 值为 p_ebp, 过程 Q 的栈帧%ebp 值为 q_ebp，假设这些值都是 32 位，则以下说法正确的是:

C

122.

A 内存地址 p_ebp 存放的整数值是 q_ebp

B 内存地址 p_ebp+4 存放的整数值是 q_ebp

C 内存地址 q_ebp 存放的整数值是 p_ebp

D 内存地址 q_ebp+4 存放的整数值是 p_ebp

ID: 9494

关于补码，下列说法错误的是

B

A 两数的补码之和（差）=两数和（差）的补码;

123.

B 任意整数的补码为其反码加 1;

C 补码使得减法运算可以用加法来实现，即用求和来代替求差;

D 补码使得数的符号位可以同数值部分作为一个整体参与运算;

ID: 9495

关于 32 位机器下 int，float，double 格式强制转换，错误的是?

A

124.

A int 转到 float,数字不会溢出，但不会被舍入。

B double 转 float，可能溢出成为正无穷或负无穷。

C double 转 float，可能会被舍入。

D double 转 int，值将会向 0 舍入。

ID: 9496

浮点运算： （3.14+1e20）-1e20 在计算机中的运算结果为( )

C

125.

A 3.14
B 10000000000
C 0
D -1

ID: 9497

浮点数 IEEE754 标准对尾数编码采用的是( )

A

126.

A 原码
B 反码
C 补码
D 移码

ID: 9498

浮点加减中的对阶的( )

A

127.

A 将较小的一个阶码调整到与较大的一个阶码相同
B 将较大的一个阶码调整到与较小的一个阶码相同
C 将被加数的阶码调整到与加数的阶码相同
D 将加数的阶码调整到与被加数的阶码相同

ID: 9499

二进制小数 1001.11 表示的浮点数的十进制表示为 （ ）

B

128.

A 8.75
B 9.75
C 1001.11
D 7.75

ID: 9500

二进制串 11010110 对应的十六进制数是( )

B

129.

A 0xx0
B 0xD6
C 0XC6
D 0Xd5

ID: 9501

130. ---

下列说法正确的是 A

A 在 C 语言中一个有符号数和一个无符号数相加得到的结果一定是一个无符号数

B 在 C 语言中一个有符号数和一个无符号数相加得到的结果可能是一个有符号数

C 在 C 语言中一个有符号数和一个无符号数相减得到的结果一定是一个有符号数

D 在 C 语言中一个有符号数和一个无符号数相减得到的结果可能是一个有符号数

ID: 9502

---

131. 请判断“并发运行是指一个进程的指令和另一个进程的指令是交错执行的。“这句话是否正确 ( ) A

A 正确

B 错误

ID: 9484

---

132. 请判断“最低有效字节在最前面的方式称为小端法，最高有效字节在最前面的方式称为大端法。“是否正确( ) B

A 错误

B 正确

ID: 9483

---

133. 当执行一个运算时，如果它的一个运算数是有符号的而另一个是无符号的，那么 C 语言会( )将( )参数强制类型转换为( )数，并假设这两个数都是非负的，来执行这个运算。 D

A 显式地 无符号 有符号

B 显示地 有符号 无符号

C 隐式地 无符号 有符号

D 隐式地 有符号 无符号

ID: 9482

---

134. 下列关于移位说法正确的是( ) A

A 逻辑左移时，高位移出，低位添 0；逻辑右移时，低位移出，高位添 0

B 逻辑右移时，高位移出，低位添 0；逻辑左移时，低位移出，高位添 0

C 逻辑左移时，低位移出，高位添 0；逻辑右移时，高位移出，低位添 0

D 无符号数的移位称为算术移位，有符号数的移位称为逻辑移位

ID: 9466

---

135. -7（8 位）的二进制补码表示正确的是 B

A 11110111

B 11111001

C 111

D 10000111

ID: 9467

---

136. 6 位二进制数最大能表示的十进制整数是( )。 B

A 64

B 63

C 32

D 31

ID: 9468

---

137. 64 位有符号除法 idivl S，所得结果商存在%eax 中，余数存在%edx 中（ ） A

A 正确

B 错误

ID: 9469

---

138. 64 位有符号除法 idivl S，所得结果商存在%edx 中，余数存在%eax 中（ ） B

A 正确

B 错误

ID: 9470

---

139. 49/16 的二进制表示( ) D

A 11.00001

B 11.01

C 11.000001

D 11.0001

ID: 9471

---

140. 32 位 linux 系统中，long 类型的字节数是 B

A 2

B 4
C 6
D 8
ID: 9472

3×4096+15×256+5×16+17 计算结果的二进制表示包含多少个 1?

B

141.

A 8
B 9
C 10
D 12
ID: 9473

汇编语言的优点不包括 D

142.

A 直接有效地控制硬件
B 生成的代码序列短小
C 运行速度快
D 编程容易
ID: 9474

---

cpu 常使用( )保存运算结果的条件代码、系统运行状态等信息

B

143.

A 程序计数器
B 程序状态（状态条件）寄存器
C 累加寄存器
D 指令寄存器
ID: 9475

1100|1010， 1001^1001， 1001&1100 分别为( )。 A

144.

A 1110 0000 1000
B 1000 1001 1000
C 1110 1001 0101
D 1000 1001 1000
ID: 9476

操作系统内核是应用程序和硬件之间的媒介。它提供三个基本的抽象。下面哪一个不是它所提供的抽象( ) D

145.

A 文件是对 I/O 设备的抽象
B 虚拟存储器是对主存和磁盘的抽象

C 进程是对处理器、主存和 I/O 设备的抽象
D 线程是对处理器、主存和 I/O 设备的抽象
ID: 9477

---

146. 请判断“数据传递、局部变量的分配和释放通过操作程序堆来实现。“是正确的吗？（ ）B
A 正确
B 错误
ID: 9478

---

147. 在某些极端要求性能的场合，我们需要对程序进行优化，关于优化，以下说法正确的是（ ）D
A 将程序整个用汇编语言改写会大大提高程序性能
B 在优化前，可以先确定哪部分代码最费时，然后对这部分代码用汇编改写，使用汇编的语句越少，程序运行越快
C 使用汇编语句虽然可以提高程序的性能，但会降低程序的可移植性，所以应该绝对避免
D 适当调整汇编指令的顺序，可以缩短程序运行的时间
ID: 9479

---

148. 请判断关于指针的说法，下面哪一个是错误的？（ ）C
A 每个指针都有一个值
B 运算符 * 用于指针的间接引用
C 将指针从一种类型强制转换成另一种类型，它的类型和值都会改变
D 数组与指针紧密联系
ID: 9480

---

149. 请判断“程序寄存器组是唯一能被所有过程共享的资源。“这句话是否正确（ ）A
A 正确
B 错误
ID: 9481

---

150. 对于源码反码和补码下列说法正确的是B
A 原码和反码不能表示 -1，补码可以表示 -1；
B 三种机器数均可表示 -1；
C 三种机器数均可表示 -1，且三种机器数的表示范围相同；
D 三种机器数均不可表示 -1。
ID: 9503

---

151. 在 C 语言中，移位运算的优先级比加法（和减法）要高。 B

A 对

B 错

ID: 9654

---

152. movb 是用来 C

A 传送字

B 传送双字

C 传送字节

D 传送地址

ID: 9674

---

153. MOV 指令 movl$0x4050 , %eax 中，源操作数类型和目的操作数类型分别为 A

A 立即数、寄存器

B 寄存器、存储器

C 存储器、寄存器

D 立即数、存储器

ID: 9675

---

154. M[R[Eb]]这种寻址方式属于 D

A 立即数寻址

B 寄存器寻址

C 绝对寻址

D 间接寻址

ID: 9676

---

155. linux 中一个源程序 hello.c 到 hello.out 的顺序为( )1 预编译　2 链接　3 编译　4 汇编 C

A 1234

B 1243

C 1342

D 1324

ID: 9677

---

156. leave 指令和下面那种处理实现的功能一致? A

A mov %ebp, %esp pop %ebp

B pop %eip

C mov %esp, %ebp pop %esp
D ret
ID: 9678

---

157. （）是一个临时的存储设备，在处理器执行程序时，用来存放程序和程序处理的数据。 D
A 外存
B 中央处理单元
C 总线
D 主存
ID: 9679

---

158. 机器级程序使用的存储器地址是（） B
A 物理地址
B 虚拟地址
C 真实地址
D 段地址
ID: 9680

---

159. （）不是用于最小程序缓冲区溢出攻击漏洞的最常见的机制之一。 A
A 限制访问形式
B 栈保护
C 随机化
D 限制哪部分存储器可以存储可执行代码
ID: 9681

---

160. 按照惯例，以下寄存器被划分为被调用者保存寄存器的是（） D
A %eax
B %edx
C %ecx
D %ebx
ID: 9682

---

161. 两个 w 位的数字相乘，机器可以用一种乘法指令来进行有符号和无符号整数的乘法（） （判断题） A
A 正确
B 错误
ID: 9683

两个数的 w 位补码之和与无符号之和有完全相同的位级表示（）（判断题） A

162.

A 正确

B 错误

ID: 9684

---

call 指令的效果是将（）入栈 A

163.

A 返回地址

B 起始地址

C 变量地址

D 上一条指令的地址

ID: 9685

栈帧的最顶端以两个指针界定，寄存器（）为帧指针。 B

164.

A %eax

B %ebp

C %esp

D %edx

ID: 9686

跳转表是一个（） C

165.

A 地址

B 链表

C 数组

D 栈

ID: 9687

以下哪一条指令可以用来检查无符号操作数的溢出（） A

166.

A CF

B ZF

C SF

D OF

ID: 9688

---

167. 堆栈指针 %esp 的内容是( ) B

A 栈顶单元内容

B 栈顶单元地址

C 栈底单元内容
D 栈底单元地址
ID: 9689

---

168. movl 传送的是( )字节 C
A 32
B 8
C 4
D 1
ID: 9673

---

169. movw 传送的是( )个字节 C
A 8
B 4
C 2
D 1
ID: 9672

---

170. OF 与 CF 的区别是 A
A OF 是溢出标志，CF 是进位标志
B OF 是零标志，CF 是溢出标志
C CF 是溢出标志，OF 是符号标志
D CF 是符号标志，OF 是进位标志
ID: 9671

---

171. （）保存着最近执行的算术或逻辑指令的状态信息。 C
A 程序计数器
B 整数寄存器
C 条件码寄存器
D 浮点寄存器
ID: 9655

---

172. 内存地址为 4 位，内存容量为（）个字节。 C
A 64
B 32
C 16
D 8
ID: 9656

173. ---

完整的整数结果不能放到数据类型的字长限制中去称为算数运算溢出。 A

A 对
B 错
ID: 9657

比例变址基址寻址 Imm(R1,R2,s)计算结果为 D

A Imm*(R1+R2+s)
B Imm*(R1+R2*s)
C Imm+R1+R2+s
D Imm+R1+R2*s
ID: 9658

174.

---

jmp *LABEL 要跳转的地址是 LABEL 地址中存的地址。 A

A 对
B 错
ID: 9659

175.

在任何时刻，程序计数器（PC）都指向主存中的 B

A 指令内容
B 指令地址
C 操作数内容
D 操作数地址
ID: 9660

176.

函数编译出来的代码会创建栈帧的原因不包括 A

A 所有的局部变量都能保存在寄存器中，且函数不会调用其他函数
B 有些局部变量是数组或者结构
C 函数用取地址操作符（&）来计算一个局部变量的地址
D 在修改一个被调用者保存寄存器之前，函数需要保存它的状态
ID: 9661

177.

IA32 架构下，栈的增长方式是 B

A 向上增长
B 向下增长
C 同时向上和向下增长
D 不确定
ID: 9662

178.

179.

x 和 y 的字节值分别为 0x66 和 0x93，则 ~x | ~y 的值为 B

A 0xFA

B 0xFD

C 0xFC

D 0xCC

ID: 9663

---

x=0xF0,经过 x>>2 算术运算后，x= C

A 0xF1

180. B 0x71

C 0xFC

D 0xFD

ID: 9664

---

x = 1011 0101 , x>>4（算术右移）后，x = B

A 0000 0101

181. B 1111 1011

C 1011 0101

D 0101 1011

ID: 9665

---

unsigned char 的最小值为 D

A 128

182. B 255

C -127

D 0

ID: 9666

在 32 位 linux 系统下，对于结构 typedef struct{char c;int a;}所占的字节数为 B

183. A 5

B 8

C 10

D 不确定

ID: 9667

184. ---

在 32 位 linux 系统下，对于结构 typedef struct{char* cp;int a;}所占的字节数为 B

A 4
B 8
C 12
D 16

ID: 9668

---

185. ret 指令从栈中弹出地址，并返回到 call 指令所在的地址 B

A 对
B 错

ID: 9669

---

186. pushl %ebp 的行为等价于以下( )两条指令 B

A subl $4, %esp movl %ebp, (%edx)
B subl $4, %esp movl %ebp, (%esp)
C subl $4, %esp movl %eax, (%esp)
D subl $4, %eax movl %ebp, (%edx)

ID: 9670

---

187. 定义数组：shor A[7]，已知 short 类型占 2 个字节，设数组的起始地址为 Xa，那么 A[2]的地址为 （ ） B

A Xa+2
B Xa+4
C Xa+6
D Xa+8

ID: 9690

---

188. 定义 2 个结构体类型：

```
struct foo1{
   float   a;
      double  b;
      float   c;
      double  d;
      short   e;
   long    f;
```

```
    short   g;
      long    h;
      char    i;
      int     j;
      char    k;
      int     l;
}

 struct foo2 {
   double  b;
      double  d;
      long    f;
      long    h;
      float   a;
      float   c;
      int     j;
      int     l;
      short   e;
      short   g;
      char    i;
      char    k;
   };
```

哪个结构体可能更节省内存？ ()

B

A foo1

B foo2

C 没有区别

D 依计算机不同而不同

ID: 9691

当执行完下列两条指令后，标志位 CF 和 SF 的值为( )。

MOV $0xC4H，%AL；ADD $0x9DH，%AL

D

189.

A 0,0

B 1,1

C 0,1

D 1,0

ID: 9692

---

190. 有符号数与无符号数运算的结果为

B

A 有符号数

B 无符号数

C 0
D 不确定

ID: 9712

---

191. 当调用 malloc 这样的 C 标准库函数时，（）可以在运行时动态的扩展和收缩，（）在程序执行期间也可以动态地扩展和收缩 B
A 堆 共享库
B 堆 栈
C 栈 内核虚拟存储器
D 栈 共享库
ID: 9713

---

192. 关于编译器优化错误的是 C
A 编译器优化依然需要程序员在高级语言编写时考虑性能问题
B 编译器的优化有时会导致最终代码与源程序码大相径庭
C 既然有不同优化，应选择最高优化级别以获取最佳性能
D 不同编译器的优化也会不同
ID: 9714

---

193. 最初的 8086 中，寄存器的特殊用途可从名字反映出来。累加器为（） A
A %ax
B %bx
C %cx
D %dx
ID: 9715

---

194. 字长相同但格式不同的两种浮点数，假设前者阶码长、尾数短，后者阶码短、尾数长，其他规定均相同，则它们可表示的数的范围和精度为( ) B
A 两者可表示的数的范围和精度相同
B 前者可表示的数的范围大但精度低
C 后者可表示的数的范围大且精度高
D 前者可表示的数的范围大且精度高
ID: 9716

---

195. 在说明语句：int *f( );中，标识符 f 代表的是( ) D
A 一个用于指向整型数据的指针变量

B 一个用于指向一维数组的行指针
C 一个用于指向函数的指针变量
D 一个返回值为指针型的函数名
ID: 9717

---

196. 在 Intel IA32 汇编中，下列哪条指令不合法？ A
A pop %eip
B pop %ebp
C mov (%esp),%ebp
D lea 0x10(%esp),%ebp
ID: 9718

---

197. 与 cmova 汇编指令等价的指令是 B
A cmovnb
B cmovnbe
C cmovnl
D cmovnge
ID: 9719

---

198. 有一个 CRT 的分辨率是 1024×768 像素，颜色数为 256 色，则刷新存储器的容量是( ) A
A 768KB
B 512KB
C 256KB
D 2MB
ID: 9720

---

199. 以下指令不改变任何条件码的是( ) C
A cmp
B test
C leal
D add
ID: 9721

---

200. 以下( )寄存器为被调用者保护寄存器。 C
A %eax
B %edx
C %ebx
D %ecx

ID: 9722

一个 8 位二进制整数采用补码表示，且由 3 个“1”和 5 个“0”组成，则最小值为( ) C

201.

A –127
B –32
C –125
D –3
ID: 9723

下面是关于汇编语言程序中使用 RET 的描述，不正确的是( ) C

202.

A 每一个子程序中允许有多条 RET 指令
B 每一个子程序结束之前一定要有一条 RET 指令
C 每一个子程序中只允许有一条 RET 指令
D 以过程形式表示的代码段，一定有 RET 指令存在
ID: 9724

下面的代码片断常常出现在库函数的编译版本中：

```
call next
next:
 popl %eax
```

203.

寄存器%eax 被设置成了什么值( ) A

A popl 这一行指令的地址
B %ebp
C %esp
D call 这一行指令的地址

ID: 9725

下面 C 语言语句中数据类型及其典型的取值范围搭配正确的是( ) A

204.

A char [-128,127]
B unsigned char [0,128]
C short [-32768,32768]
D unsigned short [0,32767]
ID: 9726

205.

下面 4 条指令中，正确的指令有（ ）条 （1）movb %ah,%sh
（2）movl %eax,$0x123
（3）movl %eax,%dx
（4）movb %si, 8(%ebp)@(a) A

A 0
B 1
C 2
D 3
ID: 9727

---

206. 下面哪项是正确的 B

A 整数一律用反码来表示
B 有符号正数的原码、反码、补码都一样
C 无符号整数有原码，但是没有反码和补码
D 有符号负数的补码是对它的反码各位取反，并在末位加 1 得到的
ID: 9711

---

207. 根据操作数特点，下面哪项是恰当的 MOV 类指令（）
_____ %eax,%ebx
_____ (%eax),%bx
_____ %eax,%ecx A

A movl movw movl
B movl movl movw
C movw movl movw
D movw movw movl
ID: 9710

---

208. 下列哪项是正确的 D

A 寄存器%eax，%ebx，%ecx 被划分为由调用者保存的寄存器
B 寄存器%edx，%esi，%edi 被划分为由被调用者保存的寄存器?
C 寄存器部分的大小必须与指令最后一个字符(b、w、l 或 q)指定的大小匹配，内存也如此
D 传送指令 MOV 的两个操作数不能都指向内存位置
ID: 9709

---

209. 在 IA32 体系结构中，当程序顺序执行时，每取一条指令语句，IP 指针增加的值是( ) D

A 1
B 2

C 4
D 由指令长度决定
ID: 9693

---

210. 存储器的一个字节表示( )位。 A
A 8
B 16
C 32
D 64
ID: 9694

---

211. 程序计数器 PC 在( )部件中。 B
A 运算器
B 控制器
C 存储器
D I/O 接口
ID: 9695

---

212. 程序计数器 PC 属于( ) B
A 运算器
B 控制器
C 存储器
D I/O 设备
ID: 9696

---

213. 程序存储器是通过( )来寻址 B
A 偏移地址
B 虚拟地址
C 物理地址
D 指令地址
ID: 9697

---

214. 插入 printf( )声明的阶段是 （） A
A 预处理阶段
B 编译阶段
C 链接阶段
D 汇编阶段
ID: 9698

215. ---

参数 x 的十六进制表示为 8，且 x 是长度 w=4 的位模式，则 x 的补码的非运算的十进制表示为( ) B

A 8

B -8

C 1

D 0

ID: 9699

---

216. 比例变址寻址 0x12(%edx, %edx, 4) 表示的操作数值为( ) D

A R[%edx]

B M[0x12 + R[%eax] * 5]

C R[0x12 + M[%edx] * 5]

D M[0x12 + R[%edx] * 5]

ID: 9700

---

217. x 和 y 的字节值分别为 0x66 和 0x93，则 x | y 的值为( ) D

A 0xF6

B 0xF4

C 0xF8

D 0xF7

ID: 9701

---

218. x 和 y 的字节值分别为 0x66 和 0x93，则 x && y 的值为( ) A

A 0x01

B 0x02

C 0x03

D 0x04

ID: 9702

---

219. x 和 y 的字节值分别为 0x66 和 0x93，则 x & y 的值为 C

A 0x03

B 0x04

C 0x02

D 0x03

ID: 9703

---

220. 下面汇编代码:

movl %eax,(%ecx)
movl (%ebx),%eax
addl $5,%eax
subl $8,%edx

下面哪项是错误的=___ B

A 第一行的目的是将%eax 的值存放到%ecx 所指向的内存地址中
B 第二行的目的是将%eax 的值存放到%ebx 中
C 第三行的目的是将%eax 的值加 5
D 第四行的目的是将%edx 的值减 8

ID: 9704

---

221. 假设在%ah 中存放的值为 0x51,%al 中存放的值为 0xe8 则在执行 cmpb %ah,%al 后。SF=____,OF=____,CF=____,ZF=____ C

A 0 0 0 0
B 0 1 0 0
C 1 0 0 0
D 1 0 1 0
ID: 9705

---

222. 8086CPU 在基址加变址的寻址方式中，变址寄存器可以为_____ D

A BX 或 CX
B CX 或 SI
C DX 或 SI
D SI 或 DI
ID: 9706

---

223. 关于跳转指令 jump，下面正确的选项是____①直接跳转：跳转目标作为指令的一部分编码
②其他条件跳转：根据 eflags 寄存器中的 CF、OF、ZF、SF 位的值，选择跳转或者继续执行下一条指令
③间接跳转：跳转目标是从寄存器或者存储器位置中读出的 D

A ①②
B ①③
C ②③
D ①②③

ID: 9707

---

224. 关于访问条件码，正确的选项为_____①条件码可以直接读取它的值 ②可根据条件码寄存器的某个组合，将一个字节设为 0 或 1 ③可直接条件跳转到程序的某个其它部分 ④可有条件的传送数据

C

A ①②

B ①②③

C ②③④

D ①②④

ID: 9708

---

225. 下列叙述正确的是 C

A 对两个无符号数进行比较采用 CMP 指令，对两个有符号数比较用 CMPS 指令

B 对两个无符号数进行比较采用 CMPS 指令，对两个有符号数比较用 CMP 指令

C 对无符号数条件转移采用 JAE/JNB 指令，对有符号数条件转移用 JGE/JNL 指令

D 对无符号数条件转移采用 JGE/JNL 指令，对有符号数条件转移用 JAE/JNB 指令

ID: 9728

---

226. 下列关于 cpu 在指令的的要求下执行的操作，说法错误的是

B

A 把一个字节或者一个字从寄存器复制到主存的某个位置

B 把两个寄存器中的内容复制到算数逻辑单元做算数操作，并将算数结果放到内存中

C 抽取指令中的一个字，将这个字复制到程序计数器

D 把一个字节或者一个字从主存复制到寄存器，覆盖寄存器原来的内容

ID: 9579

---

227. 表示法主要用于表示浮点数中的阶码 D

A 原码

B 补码

C 反码

D 移码

ID: 9599

228. ---

%esp = 0x80485C4,执行 pop %eax 后， %esp = B

A 0x80485C9

B 0x80485C8

C 0x80485C4

D 0x80485C0

ID: 9600

%edx=98765432,%al=00,则执行 movb %dh,%al 指令后， %al 为 C

229.

A 98

B 76

C 54

D 32

ID: 9601

C 语言中有定义：int x,*p;则一下正确的赋值表达式是（）。 A

230.

A *p = *&x

B *p = &x

C *p = *x

D p = x

ID: 9602

栈帧的最顶端以两个指针界定，寄存器（）为帧指针，而寄存器（）为栈指针。 B

231.

A %ebp,%eip

B %ebp,%esp

C %esp,%ebp

D %esp,%eip

ID: 9603

最常用的条件码有 CF,ZF,SF,OF。分别是（）标志、（）标志、（）标志、（）标志。 A

232.

A 进位,零,符号,溢出

B 溢出,进位,零,符号

C 零,进位,溢出,符号

D 符号,进位,溢出,零

ID: 9604

233.

计算机中，单精度浮点数据类型和双精度浮点数据类型的尾数分别用（）位和（）位表示。 D

A 22,53
B 22,52
C 23,53
D 23,52
ID: 9605

---

234. 最常见的有符号数的计算机表示方式是（）形式。 C

A 原码
B 反码
C 补码
D 移码
ID: 9606

---

235. C 语言中，“1<<2+3<<4”表达式的结果是（）。 C

A 52
B 112
C 512
D 2 的 50 次方
ID: 9607

---

236. Intel 和 AMD 提供的新硬件和以这些机器为目标的 GCC 新版本的组合，使得 x86-64 代码与为 IA32 机器生成的代码有极大的不同，下列不是 x86-64 代码的主要特征的是： B

A 指针和长整数是 64 位长。整数算术运算支持 8、16、32 和 64 位数据类型
B 通用目的寄存器组从 16 个扩展到 32 个
C 整型和指针类型的过程参数通过寄存器传递，而不是栈
D 如果有可能，条件操作用条件传送指令实现，会得到比传统分支代码更好的性能
ID: 9608

---

237. 文件本质上就是字符序列。 B

A 正确
B 错误
ID: 9609

238. ---

位于存储器层次结构中最顶部的是()。 A

A 寄存器
B 主存
C 磁盘
D 高速缓存
ID: 9610

ASCII 码中，A 和 a 对应的十进制整数值分别为（）和（）。

239. A

A 65,97
B 65,95
C 67,95
D 67,97
ID: 9611

---

32 位系统中，假设%eax 存的值为 0x123, %esp 存的值为 0x108, 则 pushl %eax 指令执行后%esp 的值为( )。 C

240.
A 0x123
B 0x108
C 0x104
D 0x112
ID: 9612

%eax=0x1234h,%ebx=5678h,执行 movb %al,%bl 后,%ebx=( )。

241. C

A 0x5678h
B 0x5612h
C 0x5634h
D 0x3478h
ID: 9613

%eax 存放的值为 8 则执行如下指令 sall $2， %eax %eax 值为( )。

242. c

A 2
B -2
C 32
D -32
ID: 9614

243. ---

是计算机中表示信息的最小单位[A]

A 位

B 字

C 字节

D 字长

ID: 9598

十进制数 2000 的十六进制数表示[B]

A 0x7CD

244. B 0x7d0

C 0x7e0

D 0x7d2

ID: 9597

---

gdb）x/17xw sum 调试命令中参数 w 表示[C]

A 单字节

B 双字节

C 四字节

D 八字节

245. ID: 9596

参数u可使用下面字符代替：

b：表示单字节

h：表示双字节

w：表示四字节

g：表示八字节

关于整数截断的规则，说法错误的是[C]

A 截断结果意义与之前不同，需要重新解释

246. B 对无符号数来说就是取模运算 mod

C 对有符号数来说就是取模运算 mod

D 有符号数截断可能会导致符号翻转

ID: 9580

“大端法或小端法存储不会影响字符串存储排列”的说法是[A]

247. A 正确的

B 错误的

ID: 9581

---

248. 下列关于操作系统说法错误的是 C

A 操作系统是应用程序和硬件之间插入的一层软件

B 操作系统具有防止硬件被失控的应用程序滥用的功能

C 操作系统具有向应用程序提供简单一致的机制来控制所有硬件设备的功能

D 操作系统通过进程、虚拟存储器和文件来实现功能

ID: 9582

---

249. 下列关于条件码的使用方法说法错误的是 C

A 根据条件码的某个组合，将一个字节设置为 0 或者 1

B 条件跳转到程序的某个其他的部分

C 条件码不可以手动设置

D 有条件地传送数据

ID: 9583

---

250. “程序寄存器组是唯一能被所有过程共享的资源”的说法是 A

A 正确的

B 错误的

ID: 9584

---

251. “寻址公式的任何一项都可以缺省”的说法是 A

A 正确的

B 错误的

ID: 9585

---

252. 关于结构体的存储要遵循的对齐规则说法错误的是 A

A 结构的首地址必须是最小元素字节数的整数倍

B 结构中每个数据的地址必须是其字节数的整数倍

C 结构的总体长度必须是最大元素字节数的整数倍

D 结构体中元素按照定义顺序放入内存

ID: 9586

---

253. 1/4 的二进制表示( ) C

A 0.02

B 0.25

C 0.01
D 0.001
ID: 9587

---

254. 0x503c-64=[B]
A 0x507c
B 0x4ffc
C 0x4efc
D 0x5eec
ID: 9588

---

255. 0x502c-0x30=[A]
A 0x4ffc
B 0x34fc
C 0x44ff
D 0xc4f1
ID: 9589

---

256. 0x502c+64=[B]
A 0x506
B 0x506c
C 0x507c
D 0x5033
ID: 9590

---

257. 0x502c+0x8=[D]
A 0x3c
B 0x55
C 0x501
D 0x5034
ID: 9591

---

258. 0x12345678 存放在采用小端存储的机器上，地址为 0x100 到 0x103，则 78 的地址为[A]
A 0x100
B 0x101
C 0x102
D 0x103
ID: 9592

0x101C + 0x8 = C

A 1024

259. B 0x1010

C 0x1024

D 0x1025

ID: 9593

0x01234567 用小端法进行存储，则它的排列顺序为 B

A 01 23 45 67

260. B 67 45 23 01

C 01 45 23 67

D 67 01 45 23

ID: 9594

---

c 语言中，~0x41 的结果是 C

A 0x41

261. B 0x71

C 0xBE

D 0xFF

ID: 9595

%eax = x,%edx = y,执行 leal 3(%eax,%edx,2),%eax 后%eax = ( )。

C

A 3*(x+2y)

262. B 3+2x+y

C x+2y+3

D 2x+2y+3

ID: 9615

%eax = 0x100,%edx = 0x3，那么 9(%eax,%edx) = ( )。 B

A 0x100

263. B 0x10C

C 0x109

D 0x103

ID: 9616

---

264. %eax = 0x1,%edx = 0x2,那么 3(%eax,%edx,2) = ( )。 D

A 0xF

B 0xE
C 0x6
D 0x8
ID: 9617

下列式子中，结果为 0 的是,其中 X 表示任意非 0 的未知数 C

265.
A X^(~X)
B X|0xFF
C !X
D ~X
ID: 9637

下列都属于计算机低级语言的是 B

266.
A 机器语言和高级语言
B 机器语言和汇编语言
C 汇编语言和高级语言
D 高级语言和数据库语言
ID: 9638

下列指令是无条件跳转 A

267.
A jmp
B je
C js
D jg
ID: 9639

完成将累加器 AL 清零，并使进位标志 CF 清零，下面错误的指令是

268.
A MOV $0x00H，%AL
B AND $0x00H，%AL
C XOR %AL，%AL
D SUB %AL，%AL
ID: 9640

条件转移指令中结果不为零（或不相等）则转移的指令是 A

269.
A jnz
B jng
C jnl
D jna

ID: 9641

---

270. 使用向偶数舍入的方式取整数，则 1.40 和 1.50 分别得到的值为 B

A 1 1
B 1 2
C 2 1
D 2 2
ID: 9642

---

271. 声明数组 int A[5][3]，数组元素的起始地址为 XA，则元素 A[3][1]的地址为 C

A XA+12
B XA+16
C XA+40
D XA+52
ID: 9643

---

272. 设有两个 5 位表示的有符号数 x = [01100],y = [00100],进行计算 x+y，结果会 D

A 负溢出
B 正常
C 无法预知
D 正溢出
ID: 9644

---

273. 设有 8 位浮点数，阶码占 4 位，尾数占 3 位，符号位占 1 位，那么在下面的位表示中，是非规格化数的是 A

A 10
B 1100
C 10001111
D 1110111
ID: 9645

---

274. 若有定义：int x,*pb;则以下正确的赋值表达式是 A

A pb=&x
B pb=x
C *pb=&x
D *pb=*x
ID: 9646

若要表示 0 ~ 999 中的任意一个十进制数，最少需( )位二进制数

D

275.

A 7
B 8
C 9
D 10
ID: 9647

若浮点数用补码表示，则判断运算结果为规格化数的方法是

C

A 阶符与数符相同，则为规格化数
276. B 小数点后第一位为 1，则为规格化
C 数符与小数点后第 1 位数字相异，则为规格化数
D 数符与小数点后第 1 位数字相同，则为规格化数
ID: 9648

若定点整数 32 位，含 1 位符号位，补码表示，则所能表示的绝对值最大的负数为 C

277.

A -2^32
B -(2^32-1)
C -2^31
D -(2^31-1)
ID: 9649

若 int len = strlen(s)，如果 s="hell"，则 len= A

A 4
278. B 5
C s 的首地址
D 0
ID: 9650

在 IA32 体系中，函数的参数是通过( )进行传递 A

A 栈
279. B CPU
C 寄存器
D 栈顶指针

ID: 9651

280. ---

字符串“\ta\017bc”的长度(不包括结束符)是 B

A 9

B 5

C 6

D 7

ID: 9652

---

281. MOVZ 类中的指令是把目的中剩余的字节填充为 0，而 MOVS 类中的指令通过符号扩展来填充，把源操作的最高位进行复制。 A

A √

B ×

ID: 9636

---

282. mov 类指令源操作数可以是立即数、寄存器或存储器操作数，目的操作数可以是寄存器或存储器操作数，但两个操作数不能同时为存储器操作数。 A

A √

B ×

ID: 9635

---

283. 268、 根据惯例，寄存器%rbx、%rbp 和%r12~%r15 被划分为被调用者保护寄存器。 A

A √

B ×

ID: 9634

---

284. %dh=CD， %eax=98765432 执行 movsbl %dh, %eax 则%eax=( )。 B

A 987654CD

B FFFFFFCD

C 000000CD

D 111111CD

ID: 9618

---

285. 在 c 语言中，计算!!0x41 的结果用十六进制表示为( )。 D

A 0X00

B 0X41
C 0X14
D 0X01
ID: 9619

---

8 位补码 10010011 等值扩展为 16 位后，其二进制表示为( )。

A

286.

A 1111111110010011
B 0000000010010011
C 1000000010010010
D 1111111101101100
ID: 9620

---

c 语言中，执行如下语句后：
int a=-25，b=25;
a=a>>3;
b=b>>3;

287.

a，b 的值分别为( )。 A

A -4,3
B -4,4
C -3,3
D -3,4
ID: 9621

当我们调用汇编器的时候，下面代码不会产生错误消息的一项是( )

D

288.

A movb $0xF, (%bl)
B movl %ax, (%esp)
C movw (%eax), 4(%esp)
D pushl $0xFF

ID: 9622

设一个栈的输入序列为 A，B，C，D，则所得到的输出序列不可能是()。 D

289.

A ABCD
B DCBA
C ACDB
D DABC
ID: 9623

---

290. 下列数据中其数值最小的是( )。 C

A 11011001B
B 75
C 037
D 2AH

ID: 9624

---

291. 在进行函数调用的过程中，有些寄存器的值只有在用户修改时才会改变，下列哪个寄存器是这样的类型? D

A %esp
B %ebp
C %ebx
D %eax

ID: 9625

---

292. 在定点二进制运算器中，减法运算一般通过( )来实现。 D

A 原码运算的二进制减法器
B 补码运算的二进制减法器
C 补码运算的十进制加法器
D 补码运算的二进制加法器

ID: 9626

---

293. 下列表述 Intel 和 ATT 格式不同的语句中错误的是 C

A Intel 代码省略了指示大小的后缀
B Intel 代码省略了寄存器名字前面的'%'符号
C 在带有多个操作数的指令情况下，列出的操作数顺序相同
D Intel 代码用不同的方式来描述内存中的位置，如是
'QWORD PTR [rbx]'而不是'(%rbx)'

ID: 9627

---

294. 跳转指令 JMP 的编码方式都是采用 PC 相对的，以目标指令的地址与紧跟在跳转指令后面那条指令的地址之间的差作为编码。 A

A √
B ×

ID: 9628

295. ---

条件码使用方式表述错误的是[C]
A 根据条件码的某种组合，将一个字节设置为 0 或者 1
B 可以条件跳转到程序的某个其他的部分
C 可以直接切换到程序中一个全新的位置
D 可以有条件的传输数据
ID: 9629

---

296. 条件转移指令 JNE 的测试条件是[A]
A ZF=0
B CF=0
C ZF=1
D CF=1
ID: 9630

---

297. 指令 pushq %rbp 的行为等价于下面两条指令:
subq $8,%rsp; movq %rbp,(%rsp)[A]
A √
B ×
ID: 9631

---

298. cltq 指令无操作数，只作用于寄存器%eax 和%rax，把%rax 符号扩展到%eax。[B]
A √
B ×
C XOR AL,DX
D SUB AX,05H
ID: 9632

299. 寄存器 rax 初始化后值为 0011223344556677，经过指令 movl $-1,%eax，寄存器 rax 中数值变成[C]
A 00112233445566FF
B 001122334455FFFF
C 00000000FFFFFFFF
D FFFFFFFFFFFFFFFF

ID: 9633

---

300. 汇编器（as）将 hello.s 翻译成（）。[C]
A C 程序
B 文本文件

C 机器语言指令
D 可执行文件
ID: 9653