一、单选题

1.寄存器的值有时是地址，有时是操作数本身，它们在形式没有区别，只有通过（  C ）才能识别它是数据还是代表地址。

A 寄存器编号   B 判断程序   C 指令操作吗或寻址方式位   D 时序信号

2. 某计算机的单字长指令16位。采用扩展操作吗方式，每个地址码4位，三地址指令15条，二地址指令8条，一地址指令127条，零地址指令（A）条

A 16   B15   C 31 D32

3 在计算机系统中，描述系统运行状态的是（D）

A 程序计数器  B 通用寄存器   C 堆栈指针   D 状态标志寄存器

二、分析题

1. 教材（袁），p139 第3题

因为该指令为相对寻址方式，当前执行时PC的内容为200，转移后为100，则说明这个相对位移量为-100，转化成相应的二进制补码形式为：10011100

2. 教材（袁），p139 第4题

（1）有效地址为120CH

（2）有效地址为12FCH

（3）有效地址为1200H

三、查阅文献和书籍,了解汇编器的作用.生成的目标文件所包含的内容.（请参考计算机组成与设计硬件与软件接口(第四版)--中文版 P83-P87以及相关文献。

，书pdf在课程资源中）

1 链接器的作用,工作步骤.

链接器有两个作用，（1）解析文件中的变量、函数名称、语句等内容生成符号表。（2）一个程序往往有多个部分组成，链接器将多个目标文件链接之后，生成一个新的可执行文件。它把所有独立汇编的机器语言程序拼接在一起

工作步骤：1、将代码和数据模块象征性地放入内存。也就是之前存在于寄存器的数据写入内存，在调用结束后恢复。指向调用点的地址或链接，允许过程返回到合适的地址 2、决定数据和指令标签的地址。3、修补内部和外部引用。

2 加载过程的简要步骤.

（1）在主程序中每次调用函数时，先依次把各参数以相反的顺序入栈；  
  然后call func\_name, 这里call要做两件事: 一是把函数的返回地址入栈，二是让指令执行指针%eip指向函数开始处。

（2）函数开始执行，首先把原来的基地址寄存器%ebp值入栈，因为在程序执行中%ebp要另作它用, 接着堆栈指针%esp的值复制给%ebp, 此后在函数执行中%ebp一直保持不变，可以由此寻址获得函数参数。

（3）开始执行函数代码。函数先要把它的局部变量保存在栈中。比如要保存一个long型数据，只要把%esp指针向下移动4个字节(因为栈增长方向是由高地址到低地址)，再根据%esp把该数据移入.

（4）函数执行要结束，要做下面几件事:  
a. 把函数的返回值存放在通用寄存器%eax中，供外部使用  
b. 把%esp指向函数开始执行的位置, 即movl %ebp,%esp  
c. 在函数返回ret之前，要还原ebx, 即popl %ebp

3  动态链接库 作用、链接和加载过程

作用：使进程可以调用不属于其[可执行代码](https://baike.baidu.com/item/%E5%8F%AF%E6%89%A7%E8%A1%8C%E4%BB%A3%E7%A0%81)的函数。函数的可执行代码位于一个 DLL 文件中，该 DLL 包含一个或多个已被编译、链接并与使用它们的进程分开存储的函数。DLL 还有助于共享数据和资源。多个应用程序可同时访问内存中单个 DLL 副本的内容。

链接和加载过程：加载程序先为新的进程创建一个虚拟地址空间，并将可执行模块映射到新进程的地址空间中。加载程序接着解析可执行模块的导入段。对导入段中列出的每个DLL，加载程序会在用户的系统中对该DLL模块进行定位，并将该DLL映射到进程的地址空间。DLL模块可以从其它DLL模块导入函数和变量，因此DLL模块可能有自己的导入段并需要将它所需要的DLL模块映射到进程的地址空间中。当加载程序将所有的DLL模块都载入并映射到进程的地址空间中后，它开始修复所有对导入符号的引用，它会检查每个模块的导入段，对导入段中列出的每个符号，加载程序会检查对应DLL的导出段，看该符号是否存在。