1. 解释中断向量

中断向量就是中断服务程序的首地址

把中断/异常与相应的处理方法对应起来。每种中断都会对应一个中断向量号，而这个向量号通过 IDT（中断描述符表）就与相应的中断处理程序对应起来了。

2. 解释中断类型码

把每个中断服务程序进行编号，这个号就代表一个中断服务程序，就是中断类型码。这个中断类型码是计算机用来查找中断向量用的

中断指令的一般格式为“INT n”，其中，n 被称为“中断类型码”

3. 解释中断向量表

中断向量表是指中断服务程序入口地址的偏移量与段基值，一个中断向量占据 4 字节空间。中断向量表是 8086 系统内存中最低端 1K 字节空间，它的作用就是按照中断类型号从小到大的顺序存储对应的中断向量，总共存储 256 个中断向量。

中断向量表在内存单元的最低处，地址空间为 00000H----003FFH(0-1024B)

这个地址正好和中断类型码有一种对应的关系：中断类型码*4(一个中断向量所占的空间)就等于这个中断向量的首地址。

4. 实模式下中断程序地址如何得到?

每一个中断向量所包含的地址以低位二字节存储偏移量，高位二字节存储段 地址；

中断类型号×4=存放中断向量的首地址；

按照实模式的寻址方式找到对应的中断处理的入口；

【根据中断类型码 n，从中断向量表中取得中断处理程序地址，取得的段地 址存入 CS，偏移量存入 IP。从而使 CPU 转入中断处理程序运行。】

5. 保护模式下中断程序地址如何得到?

在保护模式下，为每一个中断和异常定义了一个中断描述符，来说明中断和 异常服务程序的入口地址的属性

由中断描述符表取代实地址模式下的中断向量表

中断描述符含有中断处理程序地址信息

【以 IDTR 指定的中断描述符表的基地址为起始地址，用调用号 N*8 算出 偏移量，即为 N 号中断门描述符的首地址，根据中断门中的选择子和偏 移量得到中断处理程序入口。】

6. 中断向量的地址如何得到?

中断类型号×4=存放中断向量的首地址;

7. **实模式下如何根据中断向量的地址得到中断程序地址?**

段地址存入 CS，偏移量存入 IP（或者左移 4 位加偏移）。【中断向量的值即对应中断服务程序的入口地址值】

8. 解释中断描述符

中断描述符除了含有中断处理程序地址信息外，还包括许多属性和类型位。

每个中断描述符占用连续的 8 个字节

低地址的 0 和 1 两个字节是中断代码的偏移量 A15 ~ A0； 高地址的 6 和 7 两个 字节是中断代码的偏移量 A31 ~ A16;

2 和 3 两个字节是段选择符，段选择符和偏移量用来形成中断服务子程序的 入口地址;

4 和 5 两个字节称为访问权限字节，它标识该中断描述符是否有效、服务程 序的特权级和描述符的类型等信息;

I. P（present）：表示中断描述符的有效性;

II. DPL（descriptor privilege level）;

III. TYPE：指示中断描述符的不同类型

9. 保护模式下中断描述符表如何得到?

CPU 切换到保护模式之前，运行于实模式下的初始化程序必须使用 LIDT 指令 装载中断描述符表 IDT，将 IDT 基地址与段界值装入 IDTR。

10. 保护模式下中断门如何得到?

查中断描述符表以 IDTR 指定的中断描述符表的基地址为起始地址，用调用号 N×8 算出偏移量，即为 N 号中断门描述符的首地址，由此处取出中断门的 8 个字节

11. 保护模式下如何根据中断门得到中断处理程序地址?

查全局或局部描述符表根据中断门中的选择子（段选择符）和偏移量得到中断处理程序入口

12. 中断的分类，举例不同类型的中断?

A.从中断源的角度分类

（1）由计算机硬件异常或故障引起的中断，也称为内部异常中断。

（2）由程序中执行了中断指令引起的中断，也称为软中断。由程序员通过 INT 或 INT3 指令触发，通常当做 trap 处理，用处：实现系统调用。

（3）外部设备（如输入输出设备）请求引起的中断，也称为外部中断或Ⅰ／Ｏ中断。

B.主要分类

（1）由 CPU 以外的事件引起的中断

如 I/O 中断、时钟中断、控制台中断等。

（2）来自 CPU 的内部事件或程序执行中的事件引起的过程。

如由于 CPU 本身故障、程序故障和请求系统服务的指令引起的中断等。

13. 中断与异常的区别?

1）中断，是 CPU 所具备的功能。通常因为“硬件”而随机发生。

异常，是“软件”运行过程中的一种开发过程中没有考虑到的程序错误。

2）中断是 CPU 暂停当前工作，有计划地去处理其他的事情。中断的发生一般是可以预知的，处理的过程也是事先制定好的。处理中断时程序是正常运行的。

异常是 CPU 遇到了无法响应的工作，而后进入一种非正常状态。异常的出现表明程序有缺陷。

3）中断是异步的，异常是同步的。

中断是来自处理器外部的 I/O 设备的信号的结果，它不是由指令流中某条指令执行引起的，从这个意义上讲，它是异步的，是来自指令流之外的。

异常是执行当前指令流中的某条指令的结果，是来自指令流内部的，从这个意义上讲它们都是同步的

4）中断或异常的返回点

良性的如中断和 trap，只是在正常的工作流之外执行额外的操作，然后继续干没干完的活。因此处理程序完了后返回到原指令流的下一条指令，继续执行。

恶性的如 fault 和 abort，对于可修复 fault，由于是在上一条指令执行过程中发生（是由正在执行的指令引发的）的，在修复 fault 之后，会重新执行该指令；至于不可修复 fault 或 abort，则不会再返回。

5）中断是由于当前程序无关的中断信号触发的，CPU 对中断的响应是被动的，且与 CPU 模式无关。既可以发生在用户态，又可以发生在核心态。

异常是由 CPU 控制单元产生的，大部分异常发生在用户态。

14. 实模式和保护模式下的中断处理差别

保护模式下的中断处理与实模式下的中断处理最大区别在于寻找中断处理代码入口的方式

实模式直接根据中断号查中断向量表获取

保护模式要先从 IDTR 中获得中断描述符表的首地址，再根据调用号×8 得到中断描述符，再根据其中的选择子和偏移量获

15. 如何识别键盘组合键（如 Shift+a）是否还有其他解决方案?

建立扫描码的解析数组记录一个键在组合及非组合状态下的 实际值并设置缓冲区；声明了组合键中前者（ctrl，shift 等）相应变量，若出现其 make code 或 break code，设置其标志位以便与其他非组合键组合在解析数组中找到相应实际值【为 shift 设置一个变量，按下就设为 1，释放就设为 0】

16. IDT 是什么，有什么作用?

中断描述符表，存中断描述符

17. IDT 中有几种描述符?

中断描述符分为三类：任务门、中断门和自陷门

18. 异常的分类?

Fault，是一种可被更正的异常，而且一旦被更正，程序可以不失连续性地继续执行。返回地址是产生 fault 的指令。

Trap，一种在发生 trap 的指令执行之后立即被报告的异常，它也允许程序或任务不失连续性地继续执行。返回地址是产生 trap 的指令之后的那条指令。

Abort，不总是报告精确异常发生位置的异常，不允许程序或任务继续执行，而是用来报告严重错误的

外部中断的分类：

1）可屏蔽中断：

禁止响应某个中断，保证在执行一些重要的程序中不响应中断，以免造成迟缓而引起错误。

2）不可屏蔽中断

重新启动、电源故障、内存出错、总线出错等影响整个系统工作的中断是不能屏蔽的。

19. 用户态和内核态的特权级分别是多少?

用户态：3。

内核态：0。

20. 中断向量表中，每个中断有几个字节?里面的结构是什么?

4 个 Byte， 低地址 2 个 Byte 放偏移，高2 个 Byte 放段基址。

21. 中断异常共同点（至少两点），不同点（至少三点）

共同点

（1）都是程序执行过程中的强制性转移，转移到相应的处理程序。

（2）都是软件或者硬件发生了某种情形而通知处理器的行为

不同

（1）中断，是 CPU 所具备的功能。通常因为“硬件”而随机发生。异常， 是“软件”运行过程中的一种开发过程中没有考虑到的程序错误。

（2）中断是 CPU 暂停当前工作，有计划地去处理其他的事情。中断的发 生一般是可以预知的，处理的过程也是事先制定好的。处理中断时程序是正常运行的。异常是 CPU 遇

到了无法响应的工作，而后进入一种非正常状态。异常的出现表明程序有缺陷。

（3）中断是异步的，异常是同步的。

（4）中断或异常的返回点：

良性的如中断和 trap，只是在正常的工作流之外执行额外的操作，然 后继续干没干完的活。因此处理程序完了后返回到原指令流的下一 条指令，继续执行。

恶性的如 fault 和 abort，对于可修复 fault，由于是在上一条指令执 行过程中发生（是由正在执行的指令引发的）的，在修复 fault 之后，会重新执行该指令；至于不可修复 fault 或 abort，则不会再返回。

（5）中断是由于当前程序无关的中断信号触发的，CPU 对中断的响应是 被动的，且与 CPU 模式无关。既可以发生在用户态，又可以发生在核心态。异常是由 CPU 控制单元产生的，大部分异常发生在用户态。