

1. 解释中断向量

对应每个中断源设置一个向量。这些向量顺序存在主存储器的特定存储区。向量的内容是相应中断服务程序的起始地址和处理机状态字。在响应中断时，由中断系统硬件提供向量地址，处理机根据该地址取得向量，并转入相应的中断服务程序。

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。

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。

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 $\times 4$ =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。

5. 保护模式下中断程序地址如何得到？

以 IDTR 指定的中断描述符表的基地址为起始地址，用调用号 $N \times 8$ 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；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。

6. 中断向量的地址如何得到？

中断类型号 $\times 4$ =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？

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，按照实模式的寻址方式找到对应的中断处理的入口。

8. 解释中断描述符

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。长度 8 字节，2 个字节表示段选择符，4 个字节表示偏移量，2 个字节表示访问权限。

9. 保护模式下中断描述符表如何得到？

48 位的中断描述符表寄存器 IDTR 存放 IDT 的内存地址，包含 32 位基址和 16 位界限。

10. 保护模式下中断门如何得到？

以 IDTR 指定的中断描述符表的基地址为起始地址，用调用号 $N \times 8$ 算出偏移量，即为 N 号中断门描述符的首地址。

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？

按中断源分有内部异常中断、中断指令引起的中断、外部中断；按 CPU 内外分有中断和异常。

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。

13. 中断与异常的区别？

1. 中断由 CPU 以外的事件引起。异常来自 CPU 内部事件或程序执行中的事件引起的过程。
2. 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。
3. 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。
4. 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。
5. 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。因此处理程序完后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。
6. 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

见 4 和 5。

15. 如何识别键盘组合键（如 Shift+a）是否还有其他解决方案？

识别功能键的按下和弹起，记录该功能键是否被按下。

16. IDT 是什么，有什么作用？

中断描述符表，在 80x86 系列中为中断服务提供中断/陷阱描述符。

17. IDT 中有几种描述符？

1. 中断门描述符
2. 陷阱门描述符
3. 任务门描述符

18. 异常的分类？

1. 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。
2. 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。
3. 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。

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 个字节，低位二字节存储偏移量，高位二字节存储段地址。

21. 中断异常共同点（至少两点），不同点（至少三点）

1. 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。

2. 不同点：见 13。