操作系统实验（三）问答题参考

南京大学软件学院

2015.5

实验重点

本次作业重点：操作系统的中断与异常，IO 操作以及机制，实模式和保护模式下的中断异同。

1 问题清单

在整个实验的过程中，无论是编程还是查资料，请各位同学注意思考以下问题，助教检查时会从中随机抽取数个题目进行提问，根据现场作答给出分数。请注意，我们鼓励自己思考和动手实验，如果能够提供自己的思考结果并辅助以相应的实验结果进行说明，在分数评定上会酌情考虑。

1. 解释中断向量

   中断向量就是中断服务程序的首地址。

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

   存放所有的中断向量的地址空间。

4. 实模式下中断程序地址如何得到?

   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?

   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?

   调用号 N×4。

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)。

8. 解释中断描述符

   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；

9. 保护模式下中断描述符表如何得到?

   由 IDTR 指定中断描述符表的基地址

10. 保护模式下中断门如何得到?

    由 IDTR 指定中断描述符表的基地址，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？见实验二答案。

12. 中断的分类，举例不同类型的中断?

    从中断源角度可分为内部异常中断（计算机硬件一场或故障引起)、软中断（程序中执行的中断指令，如 INT、INT3)、外部中断或 I/O 中断（外部设备请求引起的中断)；外部中断可分为可屏蔽及不可屏蔽中断；

13. 中断与异常的区别?

    中断和异常，中断指由 CPU 以外的事件引起的中断（如 I/O 中断，时钟，控制台中断)，异常指来自 CPU 的内部事件或程序执行中的事件引起的过程（如 CPU 本身故障、程序故障和请求系统服务的指令引起的中断)，中断是异步的，异常时同步的

14. 实模式和保护模式下的中断处理差别

    最大区别在于寻找中断处理代码入口的方式——实模式下，中断处理程序的入口地址称为“中断向量”，所有的“中断向量”存储在一个“中断向量表”中；而保护模式下，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；

15. 如何识别键盘组合键（如 Shift+a）是否还有其他解决方案?

    orange‘s 中，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

16. IDT 是什么，有什么作用?

    IDT（Interrupt Descriptor Table），中断描述符表，作用是将每一个中断向量和一个描述符对应起来。

17. IDT 中有哪几种描述符?

    中断门描述符，陷阱门描述符，任务门描述符。

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 fault 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。

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 Bytes 低地址两个 Byte 放偏移高地址两个 Byte 放段描述符。

21. 中断异常共同点 (至少两点)，不同点 (至少三点) 相同点：

    (a) 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。

    (b) 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:

    (a) 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。

    (b) 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。

    (c) 中断是异步的，异常是同步的。

(d) 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。

(e) 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。

2 参考资料

1. 《Orange'S: 一个操作系统的实现》