# 中断处理程序设计

## 一.键盘中断处理程序

1.如果用户敲键盘，则IRQ1引脚有信号

(1)如果中断控制器允许（没有屏蔽）IRQ1，则INTR有信号，同时传出中断类型号9（键盘中断）

(2)如果开中断（没有屏蔽外部中断），则在执行当前指令后响应，进入键盘中断处理程序（9号中断处理程序）



2.键盘中断处理程序的功能

(1)从键盘接口取得所按键的扫描码，然后根据扫描码判定所按的键并作相应的处理。

(2)具体而言：把字符键的扫描码和对应的ASCII码存到键盘缓冲区；

(3)把功能键的扫描码存到键盘缓冲区；记录下控制键和双态键的状态；

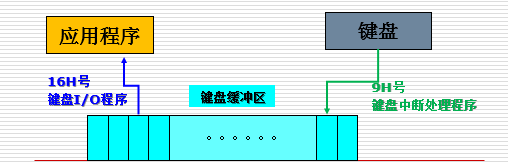
(4)直接处理特殊请求键。

3.键盘三者（缓冲区、中断处理程序、I/O程序）的关系

仓 库：键盘缓冲区（内存某个区域）

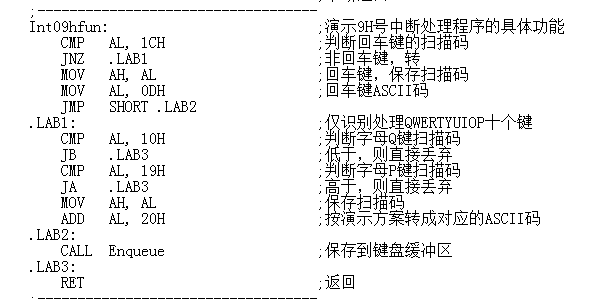
生产者：键盘中断处理程序（9H号中断处理程序）

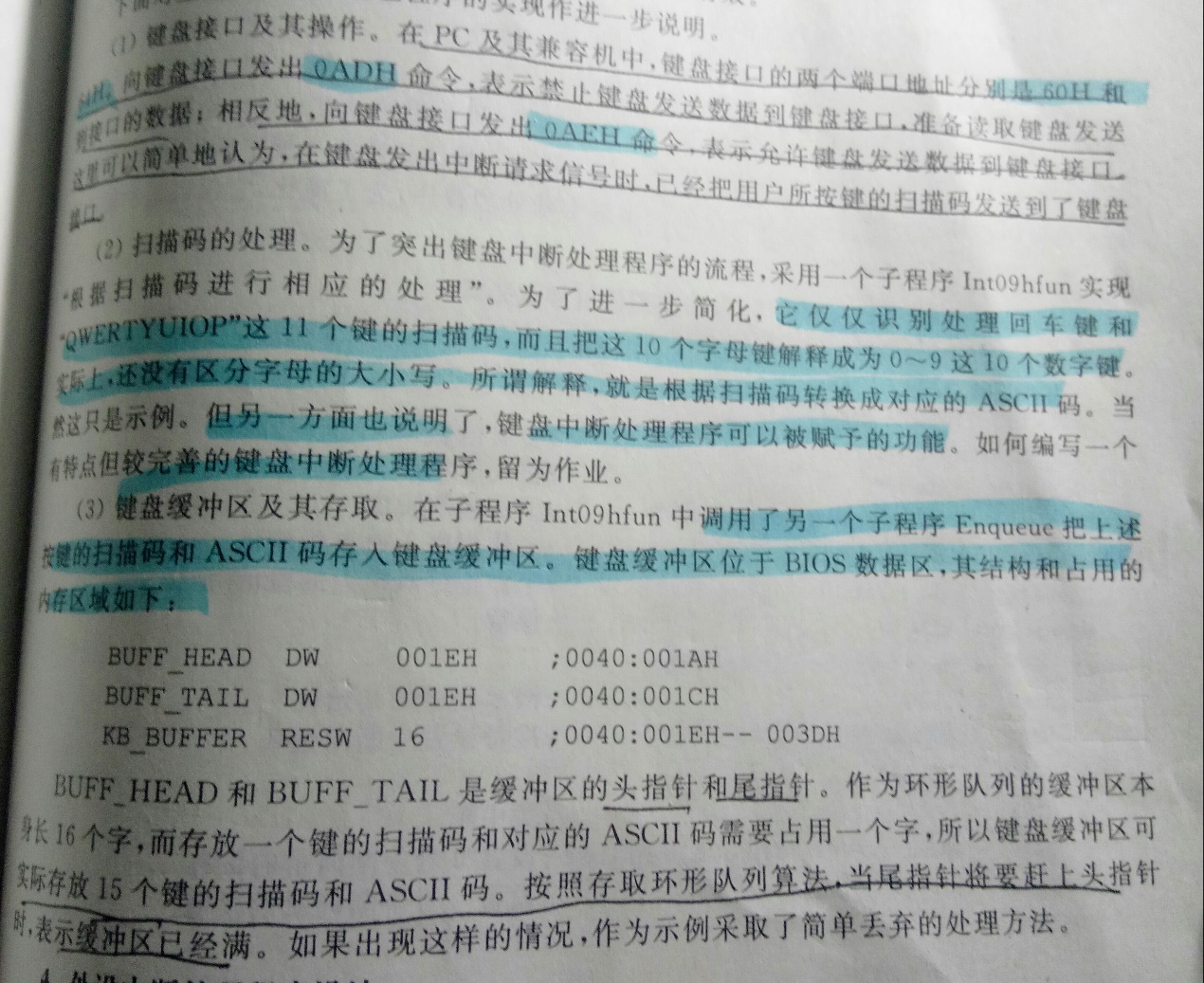
消费者：键盘I/O程序（16H号中断处理程序）



4.程序说明

(1)第一部分是初始化和演示处理：首先设置9号中断向量，使其指向新的键盘中断处理程序；然后接受用户按键，并显示所得字符，直到当用户按回车键为止。

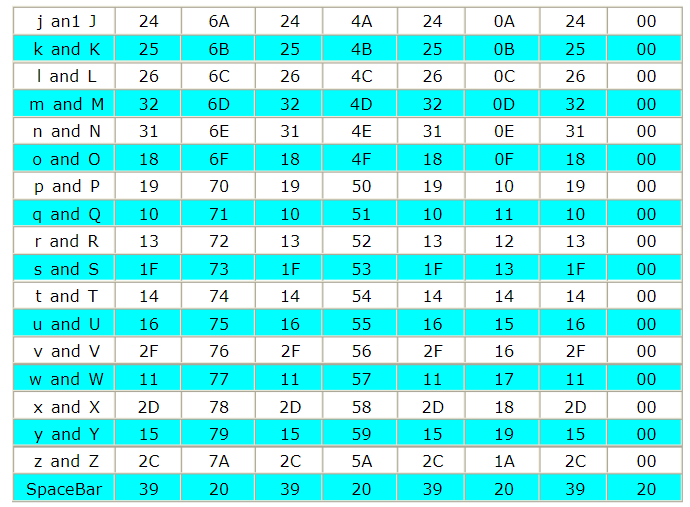
(2)第二部分是新的简化了的键盘中断处理程序。它从键盘接口取得用户所按键的扫描码，并根据扫描码进行相应的处理，在完成处理工作后，通知中断控制器8259A中断结束，最后中断返回。

(3) 

5.演示效果：

只有qwertyuiop这十个键有效，对应0到9十个数字键





当用户按回车键时，演示结束

用户按其它键，被丢弃

6.外设中断处理程序的主要步骤如下：

(1)必须保护现场

(2)尽快完成中断处理

(3)恢复现场

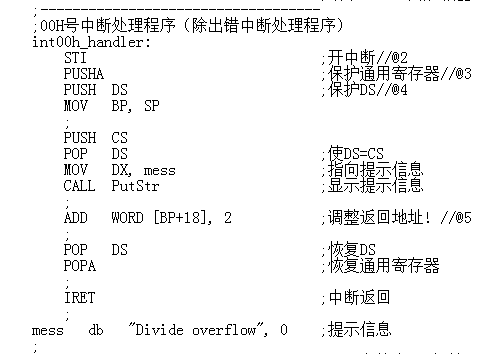
(4)通知中断控制器中断已经结束

(5)利用IRET指令实现中断返回

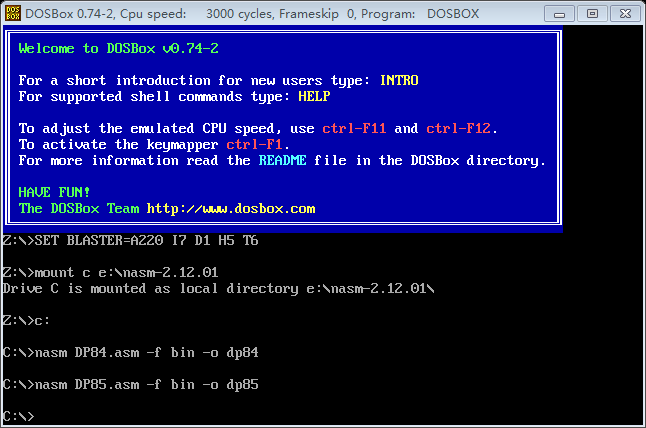
## 二.除法出错中断处理程序

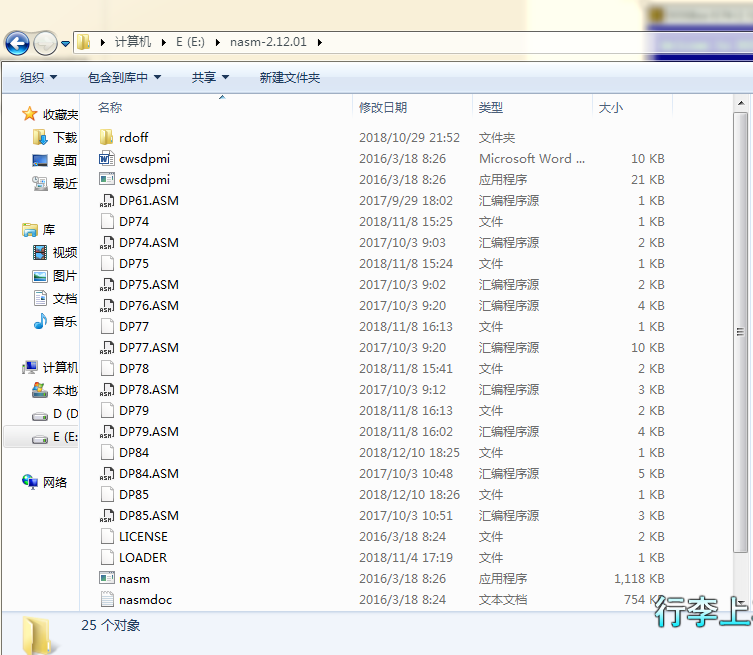
1.在执行除法指令时，如除数为0或者商超过了规定的范围，导致除法出错中断。除法出错中断类型号规定为0。

在转0号（除法出错）中断处理程序时，压入堆栈的中断返回地址是引起除法出错的指令地址

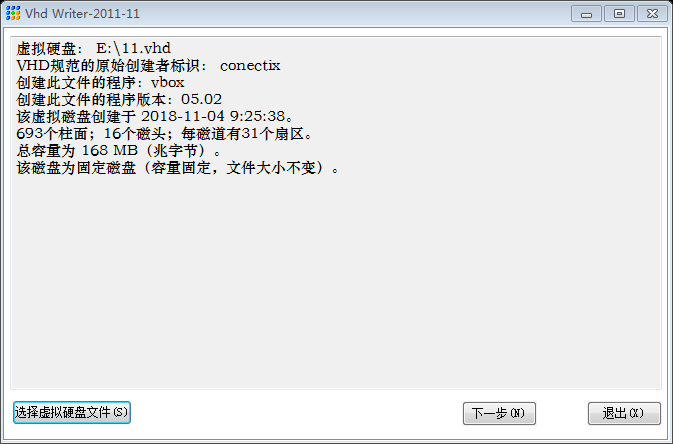
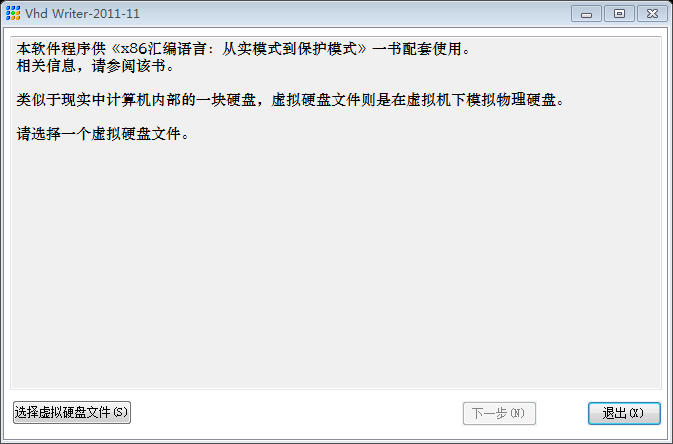
2.第一部分包含初始化和引起除法出错的指令。初始化的工作为：设置0号中断向量，使其指向除法出错中断处理程序。第二部分是除法出错中断处理程序。为了简化，它仅仅显示提示信息。

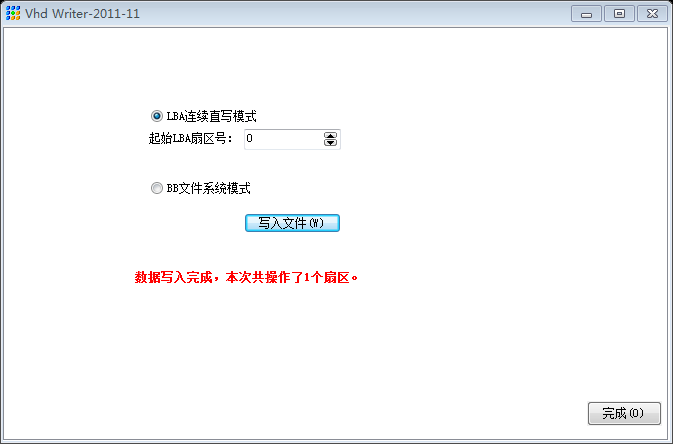
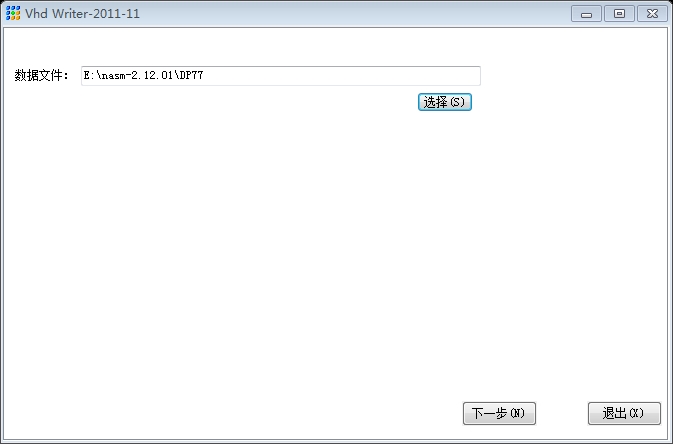
## 三.实验过程

1.将文件转换为纯二进制文件

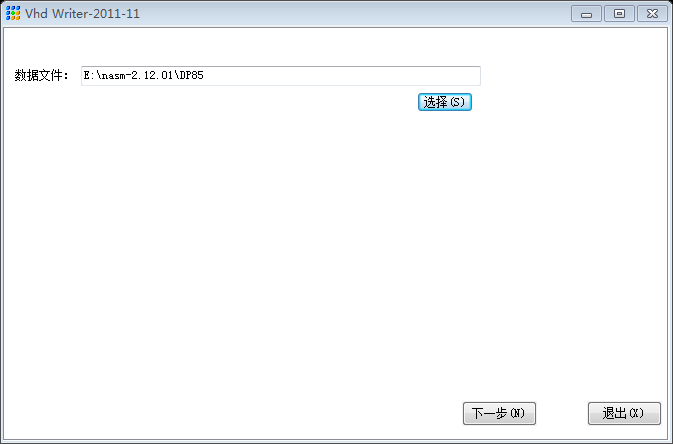
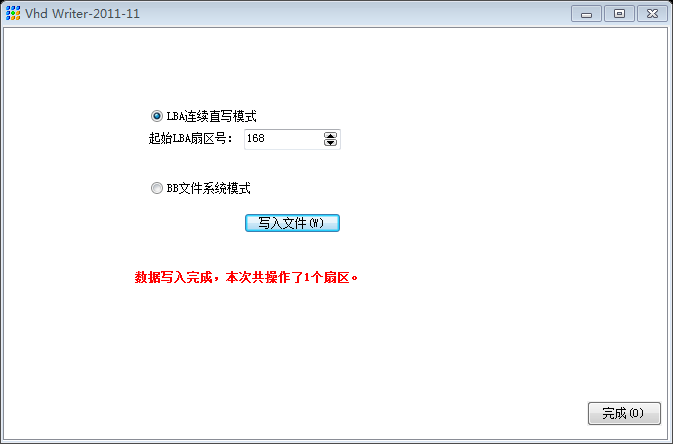
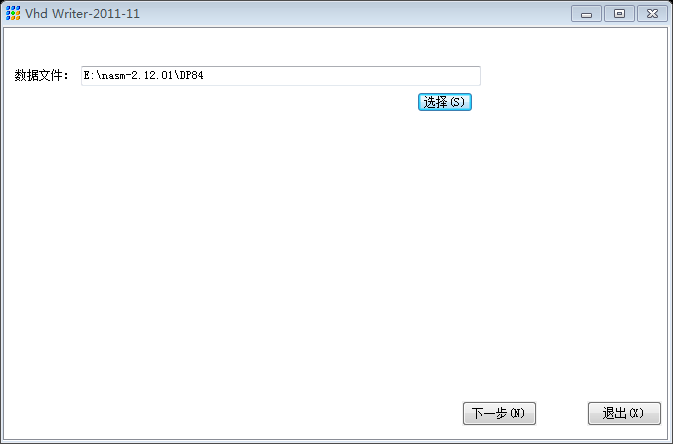


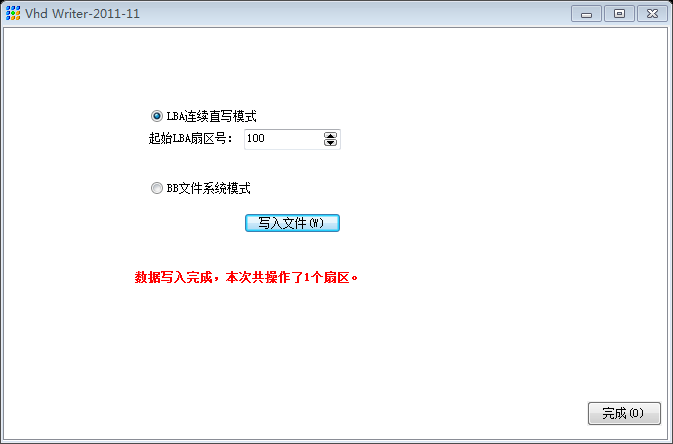
2.利用前面的dp77.asm作为加载引导程序，写入0扇区





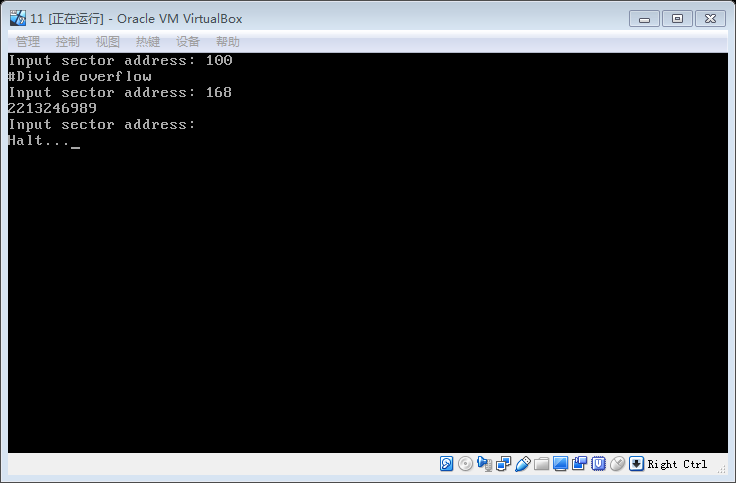
3.将dp84.asm和dp85.asm分别写入168 和100 扇区





4.输出结果：

输入相应的扇区地址，可以得到键盘中断处理程序和除法溢出中断处理程序



5.将我自己改的DP86.asm二进制文件 写入170扇区，则运行结果图如下

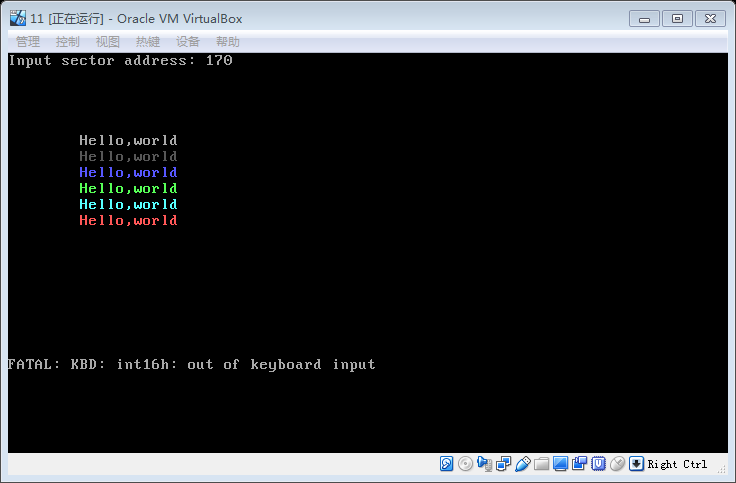
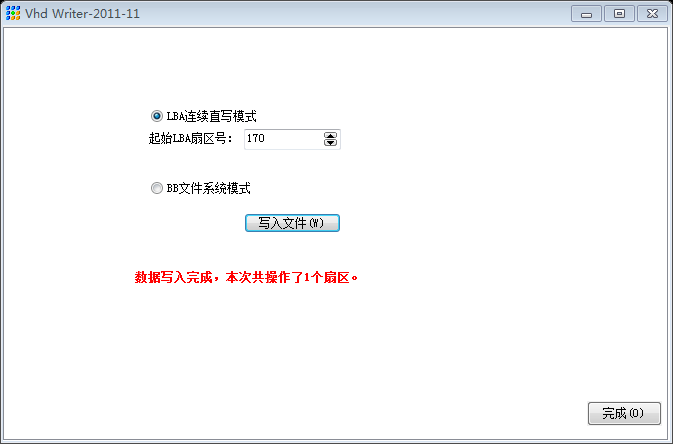
出现错误。则是出现嵌套错误。原因如下：

（1）CPU在执行中断处理程序时，又发生中断，这种情况被称为中断嵌套。

（2）在中断处理过程中，发生内部中断。

（3）在中断处理过程中，发生非屏蔽中断。

（4）如果在中断处理程序中使用了开中断指令，也就可能会发生可屏蔽中断引起的中断嵌套。



## 四.心得体会

通过此次的实验,让我对内部中断和外部中断有了一定的认识.

中断是一种使CPU挂起正在执行的程序而转去处理特殊事件的操作。中断指暂停执行当前程序，切换执行处理特殊事件的服务程序。把这样的处理特殊事件的服务程序称为中断处理程序，它就是响应处理中断的程序。

由发生在CPU内部的某个事件引起的中断被称为内部中断。由于内部中断是CPU在执行某些指令时产生，所以也称之为软件中断。内部中断的特点：不需要CPU外部硬件的支持；不受中断允许标志IF的控制。

由发生在CPU外部的某个事件引起的中断被称为外部中断。如，输入/输出设备等引起的中断就是外部中断。外部中断以完全随机的方式中断现行程序。此次的键盘中断,我们不止可以操作QWERTYUIOP这几个字母，也可以改变其他字母与其属性。我们可以依照书上的进行改写、

在实验时，当有多个中断时，我们需要识别中断的优先级。此次，我的出现了错误，我自己上网搜了也没有找到答案，可能是嵌套出错了。