

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验六 输入输出与中断处理程序设计**

**实验时间： 2016-4-26，14：00-17：30 实验地点： 南一楼804室74号实验台**

**指导教师： 张勇 专业班级：计算机科学与技术201409班**

**学 号： U201414813 姓 名： 唐礼威**

**同组学生： 无 报告日期： 2016年 5 月 2日**

**原创性声明**

  本人郑重声明：本报告的内容由本人独立完成，有关观点、方法、数据和文献等的引用已经在文中指出。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品或成果，不存在剽窃、抄袭行为。

特此声明！

学生签名：

日期：

成绩评定

|  |  |  |
| --- | --- | --- |
| 实验完成质量得分（70分）（实验步骤清晰详细深入，实验记录真实完整等） | 报告撰写质量得分（30分）（报告规范、完整、通顺、详实等） | 总成绩（100分） |
|  |  |  |

指导教师签字：

                    日期：

**目录**

[1 实验目的与要求 1](#_Toc451727928)

[2 实验内容 1](#_Toc451727929)

[3 实验过程 2](#_Toc451727930)

[3.1 任务1 2](#_Toc451727931)

[3.1.1 设计思想及存储单元分配 2](#_Toc451727932)

[3.1.2 流程图 2](#_Toc451727933)

[3.1.3 源程序 2](#_Toc451727934)

[3.1.4 实验步骤 3](#_Toc451727935)

[3.1.5 实验记录 3](#_Toc451727936)

[3.2 任务2 3](#_Toc451727937)

[3.2.1 设计思想及存储单元分配 3](#_Toc451727938)

[3.2.2 流程图 4](#_Toc451727939)

[3.2.3 源程序 4](#_Toc451727940)

[3.2.4 实验步骤 6](#_Toc451727941)

[3.2.5 实验记录 6](#_Toc451727942)

[3.3 任务3 7](#_Toc451727943)

[3.3.1 设计思想及存储单元分配 7](#_Toc451727944)

[3.3.2 流程图 7](#_Toc451727945)

[3.3.3 源程序 7](#_Toc451727946)

[3.3.4 实验步骤 9](#_Toc451727947)

[3.3.5 实验记录 9](#_Toc451727948)

[4 体会 9](#_Toc451727949)

[参考文献 10](#_Toc451727950)

# 实验目的与要求

(1) 掌握中断矢量表的概念；

(2） 熟悉I/O访问，BIOS功能调用方法；

(3) 掌握实方式下中断处理程序的编制与调试方法。

# 实验内容

任务1：用三种方式获取中断类型码10H对应的中断处理程序的入口地址。

要求：首先要进入虚拟机状态，然后

（1） 直接运行调试工具（TD.EXE），观察中断矢量表中的信息。

（2） 编写程序，用 DOS系统功能调用方式获取，观察功能调用相应的出口参数与“（1）”看到的结果是否相同 （使用TD观看出口参数即可）。

（3） 编写程序，直接读取相应内存单元，观察读到的数据与“（1）”看到的结果是否相同 （使用TD观看程序的执行结果即可）。

任务2：编写一个接管键盘中断的中断服务程序并驻留内存，要求在程序返回DOS操作系统后，键盘的按键A变成了按键B，而B变成了A。

要求：

（1）在 DOS虚拟机或DOS窗口下执行程序，中断服务程序驻留内存。

（2）在DOS命令行下按键A，屏幕显示为B，按B时屏幕显示为A。执行TD，在代码区输入指令“MOV AX,0”看是否能发生变化。

（3）选作：另外编写一个中断服务程序的卸载程序，将键盘中断服务程序恢复到原来的状态（也就是还原中断矢量表的信息）。

任务3：读取CMOS内指定单元的信息，按照16进制形式显示在屏幕上。

要求：

（1） 先输入待读取的CMOS内部单元的地址编号（可以只处理编号小于10的地址单元）。再使用IN/OUT指令，读取CMOS内的指定单元的信息。

（2） 将读取的信息用16进制的形式显示在屏幕上。若是时间信息，可以人工判断一下是否正确

# 实验过程

## 任务1

### 设计思想及存储单元分配

用三种方式获取中断类型码10H对应的中断处理程序的入口地址。

1.直接在td中观察即可。

2.系统功能调用35H，入口参数为AL=10H,取中断信息。

3.直接把主存中的DS:[40H]→IP,DS:[42H]→CS赋值给寄存器BX和CX。

### 流程图

无

### 源程序

1.程序名renwu61

.386

CODE SEGMENT USE16

ASSUME CS:CODE

START: XOR AX,AX

MOV DS,AX

MOV AX,3510H

INT 21H

MOV AH,4CH

INT 21H

CODE ENDS

END START

2.程序名renwu62

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

START: XOR AX,AX

MOV DS,AX

XOR BX,BX

XOR CX,CX

MOV BX,DS:[40H]

MOV CX,DS:[42H]

MOV AH,4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1.直接运行td，置数据区为当前区执行Goto，在地址输入窗口输:40H{（0:[10\*4])->IP,(0:[10\*4+2]->CS}读取连续的4个字节，前两个为偏移地址，后两个为段值。然后置代码区为当前区。输入刚才得到的值，得到对应的程序。

2.将程序renwu61编译、链接，调试直到没有错误。运行td renwu61，在td中单步调试，观察es与bx的改变情况；

3.将程序renwu62编译、链接，调试直到没有错误。运行td renwu62，在td中单步调试，观察寄存器的值。

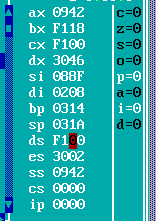
### 实验记录

（包括实验条件、输入/输出、错误和修改等信息的记录）

1.实验环境条件： 16M内存；WINDOWS XP命令行窗口；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2.方法一，直接观察ds段，可以观察到从0:40h开始低2个字节为0F96，高2个字节为3002；

3.方法二，调用35号系统功能，在TD内观察到es为3002，bx为0F96；



4.方法三，将0:40h开始的低2个字节移至bx，高2个字节移至cx，得到cx为3002，bx为0F96；

5.比较上述三种方法得到的结果，发现它们一致，说明以上方法实施得到的结果准确无误

## 任务2

### 设计思想及存储单元分配

设计思想：

修改键盘驱动中断程序(int 16h)，将中断矢量表0:58h开始的4个字节修改为新的中断程序的偏移地址和段首址，并保存号旧的键盘驱动中断程序的偏移地址和段首址。

在新的中断程序中，先判断用户调用的是否为键盘驱动中断程序的读入键盘功能，若不是，则直接调用旧的键盘驱动中断程序;若是，则对输入进行处理。

实际上我们改变出口参数中AL的值即是键入字符的ASCII码即可实现。

### 流程图



### 源程序

程序名renwu63

.386

DATA SEGMENT USE16

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE,SS:STACK

OLD\_INT DW ?,?

NEW16H:CMP AH,00H

JE ATB

CMP AH,10H

JE ATB

JMP DWORD PTR OLD\_INT

ATB:

PUSHF

CALL DWORD PTR OLD\_INT

CMP AL,41H ;比较输入的ascii码

JNZ NEXT1 ;

MOV AL,42H

JMP QUIT

NEXT1: CMP AL,42H

JNZ NEXT2

MOV AL,41H

JMP QUIT

NEXT2: CMP AL,61H

JNZ NEXT3

MOV AL,62H

JMP QUIT

NEXT3: CMP AL,62H

JNZ QUIT

MOV AL,61H

QUIT: IRET

START: XOR AX,AX

MOV DS,AX

MOV AX,DS:[16H\*4]

MOV OLD\_INT,AX ;保存偏移部分

MOV AX,DS:[16H\*4+2]

MOV OLD\_INT+2,AX ;保存段值

CLI

MOV WORD PTR DS:[16H\*4],OFFSET NEW16H

MOV DS:[16H\*4+2],CS

STI

MOV DX,OFFSET START+15

SHR DX,4

ADD DX,10H

MOV AL,0

MOV AH,31H

INT 21H

CODE ENDS

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

END START

### 实验步骤

1. 使用编辑程序EDIT.EXE录入源程序renwu63，编译，链接。

2. 观察提示信息，若出错，则用编辑程序修改错误，修改后再重新编译。

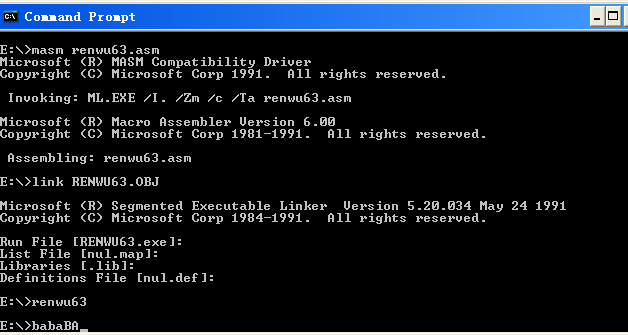
3. 执行该程序输入a/A,观察是否输出b/B;输入b/B,观察是否输出a/A。

### 实验记录

1.实验环境条件：16M内存；WINDOWS XP命令行窗口；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2. 在键盘键入a/A，发现屏幕回显b/B;在键盘键入b/B，发现屏幕回显a/A，证明程序编写得准确无误。

当输入ababAB，输出：



3.通过编写代码发现，将旧的中断程序的偏移地址与首地址保存下来后，可有有两种方式调用旧的中断程序:

Jmp dword ptr old\_int;

Call dword ptr old\_int;

前一种方式，可直接结束新的(不返回至)中断程序;后一种方式，可返回至新的中断程序。

## 任务3

### 设计思想及存储单元分配

1号系统功能调用读入字符，将用户输入转化为数字，作为信息的偏移地址。然后调用in 指令，从指定端口取出数据即可。

### 流程图



### 源程序

程序名renwu64

.386

data segment use16

buf db 2

data ends

stack segment use16 stack

db 200 dup(0)

stack ends

code segment use16

assume ds:data,ss:stack,cs:code

start:

mov ax, data

mov ds, ax

lopa: mov ah, 1 ;读入输入

int 21h

sub al, 30h ;转化成十进制0~9

out 70h, al ;取出信息

in al, 71h

mov ah, al

and al, 0fh

shr ah, 4

add ax, 3030h ;转化为ascll码

xchg ah, al

mov bx, ax

lea si, buf

mov word ptr [si+1], bx

mov dl, 0ah ;输出换行

mov ah, 2

int 21h

mov dl, 0dh

mov ah, 2

int 21h

mov dl, bl ;输出信息

mov ah, 2

int 21h

mov dl, bh

mov ah, 2

int 21h

mov dl, 0ah ;输出换行

mov ah, 2

int 21h

mov dl, 0dh

mov ah, 2

int 21h

jmp lopa ;循环

mov ah, 4ch

int 21h

code ends

end start

### 实验步骤

1. 录入源程序renwu64，编译，链接。

2. 观察提示信息，若出错，则用编辑程序修改错误，修改后再重新编译。

3. 执行该程序，‘0’~‘9’，然后观察屏幕输出信息，验证是否正确。

### 实验记录

1.实验环境条件：16M内存；WINDOWS XP命令行窗口；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2.取出的信息按照BCD编码进行压缩，分为两部分，分别存储在AL，AH。比如说当前小时信息是19，则1存储在AH，为0x01,9存储在AL,为0x09。

3.执行程序发现有时输出0x0，屏幕上一次输出当前的时间信息，包括年、月、日、星期、小时、分钟、秒。



# 体会

通过这次上机实验，明白了IN/OUT指令的工作原理，对中断向量表有了更进一步的认识，知道了如何去识别中断向量表，如何获取中断程序的段首址与偏移地址，如何扩展原有中断程序的功能等等，对今后的编程实践会有很多帮助。

# 参考文献

[1] 王元珍等.80x86汇编语言程序设计.版本(第1版)

[2] 王晓虹等.汇编语言程序设计教程.版本(第1版)