

**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： 2016-4-26，14：30-17：30 实验地点： 南一楼804室56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： U201414797 姓 名： 张丹朱**

**同组学生： 无 报告日期： 2016年 4 月 29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](#_Toc451543807)

[2 实验内容 1](#_Toc451543808)

[3 实验过程 2](#_Toc451543809)

[3.1 任务1 2](#_Toc451543810)

[3.1.1 设计思想及存储单元分配 2](#_Toc451543811)

[3.1.2 流程图 2](#_Toc451543812)

[3.1.3 源程序 2](#_Toc451543813)

[3.1.4 实验步骤 2](#_Toc451543814)

[3.1.5 实验记录 2](#_Toc451543815)

[3.2 任务2 3](#_Toc451543816)

[3.2.1 设计思想及存储单元分配 3](#_Toc451543817)

[3.2.2 流程图 3](#_Toc451543818)

[3.2.3 源程序 4](#_Toc451543819)

[3.2.4 实验步骤 5](#_Toc451543820)

[3.2.5 实验记录 5](#_Toc451543821)

[3.3 任务3 5](#_Toc451543822)

[3.3.1 设计思想及存储单元分配 5](#_Toc451543823)

[3.3.2 流程图 5](#_Toc451543824)

[3.3.3 源程序 6](#_Toc451543825)

[3.3.4 实验步骤 7](#_Toc451543826)

[3.3.5 实验记录 8](#_Toc451543827)

[4 体会 8](#_Toc451543828)

[参考文献 9](#_Toc451543829)

# 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

1. 直接在td中观察0000:0040处存放的内容
2. 用35号调用查看类型码10H对应的入口地址
3. 编程直接读取0：[10H \* 4]处内容

1.寄存器分配

AX：功能调用等

### 流程图

无

### 源程序

.386

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

START: XOR AX, AX

MOV DS, AX

MOV AX, 3510H

INT 21H

MOV DX, DS:[10H \* 4]

MOV AH, 4CH

INT 21H

CODE ENDS

END START

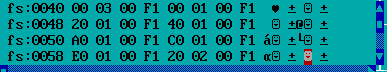
### 实验步骤

1. 使用记事本编写源代码，另存为lab6-1.asm
2. 用MASM汇编源程序：MASM lab6-1，获得LAB6-1.OBJ。
3. 用LINK链接：LINK LAB6-1;，获得LAB6-1.EXE
4. 用td调试，观察相应内存单元或寄存器的值：
5. 在数据段区域右键选GOTO，输入0000:0040观察存储单元的值
6. 单步执行程序，观察相应出口寄存器中的值

### 实验记录

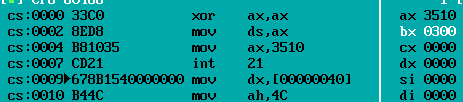
1、实验环境条件：DOSBOX0.73; MASM.EXE 6.0; LINK.EXE 5.2; TD.EXE 5.0。

2、在数据段区域右键选GOTO，输入0000:0040，观察到存储单元的内容：



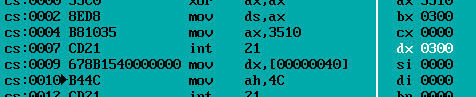
为 00 03 即 03 00

3、按F8单步执行，在35号调用后观察出口参数（ebx）的值，如下：



为 03 00 与上一方法相同

4、继续执行下一步，直接将0000:0040的内容存入dx，获得如下：



为 03 00 与上两步相同

## 任务2

### 设计思想及存储单元分配

首先调用原中断，获取返回的ASCLL码（在AL中），若为41（‘A’），则换成42（‘B’），反之将42换成41，其余则不变。

1.存储单元分配

OLD\_INT：字存储区,用于存放原中断信息

2.寄存器分配

DX：临时寄存器。

AL：存放输入字符的ASCLL码

### 流程图

先保存原中断矢量信息，再将新的中断入口地址存入中断矢量表相应位置（此一步流程图略），其中新中断的流程图如下图3.2所示：

开始

PUSHF

CALL DWORD PTR OLD\_INT

AL==41

AL==41

MOV AL, 42

MOV AL, 42

IRET

结束

N

N

Y

Y

图3.2 新中断流程图

### 源程序

.386

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

OLD\_INT DW ?,?

;新中断处理程序（“INT 16H”）的代码

NEW16H: PUSHF

CALL DWORD PTR OLD\_INT

CMP AL, 41H

JE ATB

CMP AL, 42H

JE BTA

LAST: IRET

ATB: MOV AL, 42H

JMP LAST

BTA: MOV AL, 41H

JMP LAST

;初始化（安装新中断矢量并常驻内存）程序

START: XOR AX, AX

MOV DS, AX

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

MOV OLD\_INT, AX

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

MOV OLD\_INT+2, AX

CLI

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

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

STI ;开中断

MOV DX, OFFSET START+15

SHR DX, 4

ADD DX, 10H

MOV AL, 0

MOV AH, 31H

INT 21H

CODE ENDS

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

END START

### 实验步骤

1、使用记事本编写源代码，另存为lab6-2.asm

2、用MASM汇编源程序：MASM lab6-2，获得LAB6-2.OBJ。

3、用LINK链接：LINK LAB6-2;，获得LAB6-2.EXE

4、返回DOS命令行，执行LAB6-2.EXE

5、依次按键ABCDEF

### 实验记录

1、实验环境条件：DOSBOX0.73; MASM.EXE 6.0; LINK.EXE 5.2; TD.EXE 5.0。

2、编译等都正常通过

3、执行程序

4、依次按键ABCDEF，得到：BACDEF,符合预期。

## 任务3

### 设计思想及存储单元分配

将数据地址传入端口70H，然后将读取71H端口内容并转换成16进制输出

1.寄存器分配

AX、DX：临时寄存器。

### 流程图

如下图3.3所示：

开始

一号调用输入数字字符（0-9）

AL-‘0’ -> AL

将AL值送入70H端口

从71端口获取内容并转换成16进制（调用书上RADIX函数）输出

结束

图3.3 CMOS调用流程图

### 源程序

.386

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

BUF DB 10 DUP(0)

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK, DS:DATA

;子程序名：RADIX

;功能：将EAX中的32位无符号二进制数转换成P进制数（16位段）

;入口参数：

;EAX——存放待转换的32位无符号二进制数

;EBX——存放要转换数制的基数

;SI——存放转换后的P进制ASCLL码数字串的字节缓冲区首址

;出口参数：

;所求P进制ASCLL码数字串按高位在前、低位在后的顺序存放在以SI为指针的字节缓冲区中

;SI——指向字节缓冲区中最后一个ASCLL码的下一个字节

;所使用的寄存器：

;CX——P进制数字入栈、出栈时的计数器

;EDX——做除法时存放被除数高位或余数

RADIX PROC

PUSH CX

PUSH EDX ;保护现场

XOR CX, CX ;计数器清零

LOP1: XOR EDX, EDX

DIV EBX

PUSH DX

INC CX

OR EAX,EAX ;若(EAX)!=0,跳转到LOP1

JNZ LOP1

LOP2: POP AX

CMP AL, 10

JB L1

ADD AL, 7

L1: ADD AL,30H

MOV [SI], AL

INC SI

LOOP LOP2

MOV BYTE PTR [SI], 0AH

INC SI

MOV BYTE PTR [SI], 0DH

INC SI

MOV BYTE PTR [SI], '$' ;写入结束符

INC SI

POP EDX

POP CX

RET

RADIX ENDP

START: MOV AH, 1

INT 21H

SUB AL, '0' ; 将输入字符转化为数字

MOV AH, 0

MOV DX, 70H ;将端口号送入dx

OUT DX, AX

XOR EAX, EAX

IN AX, 71H ; 读取端口信息至al

MOV EBX, 10H

LEA SI, BUF

CALL RADIX ; 将端口信息化为16进制数字串

MOV DL, ' ' ; 输出16进制数字串

MOV AH, 2H

INT 21H

MOV DL, '0'

MOV AH, 2H

INT 21H

MOV DL, 'x'

MOV AH, 2H

INT 21H

LEA DX, BUF

MOV AH, 9H

INT 21H

FINISH: MOV AH, 4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 使用记事本编写源代码，另存为lab6-3.asm。
2. 用MASM汇编源程序：MASM lab6-3，获得LAB6-3.OBJ。
3. 用LINK链接：LINK LAB6-3;，获得LAB6-3.EXE

### 实验记录

1、实验环境条件：DOSBOX0.73; MASM.EXE 6.0; LINK.EXE 5.2; TD.EXE 5.0。

2、编译等都正常通过

3、执行程序，输入4,2,0进行测试，获得当前时间，如下图3.3-2所示：

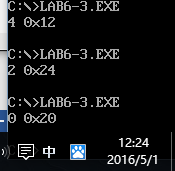


图3.3-2 测试结果图

# 体会

本次实验主要熟悉了中断矢量表。学会了用不同的方法查看中断矢量内容，并能够修改中断矢量表并驻留内存。明白了中断调用的原理。熟悉了I/O访问及BIOS功能调用方法。

总体来说本次实验较为简单。

# 参考文献

[1] 王元珍,曹忠升,韩宗芬.80X86汇编语言程序设计.第1版.武汉:华中科技大学,2005年4月：281-302