

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 中断与反跟踪**

**实验时间： 2018-5-4，18:30-21:50 实验地点： 南一楼804室xx号实验台**

**指导教师： 李海波 专业班级：计算机科学与技术201607班**

**学 号： U201614700 姓 名： 王亚宁**

**同组学生： 无 报告日期： 2018年5月6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**

[实验目的与要求 1](#_Toc515289751)

[实验内容 1](#_Toc515289752)

[实验过程 1](#_Toc515289753)

[任务1 1](#_Toc515289754)

[1. 设计思想及存储单元分配 1](#_Toc515289755)

[2. 流程图 1](#_Toc515289756)

[3. 源代码 3](#_Toc515289757)

[4. 实验步骤 4](#_Toc515289758)

[5. 实验记录 4](#_Toc515289759)

[任务2 6](#_Toc515289760)

[1. 设计思想及存储单元分配 6](#_Toc515289761)

[2. 流程图 6](#_Toc515289762)

[3. 源代码 6](#_Toc515289763)

[4. 实验步骤 7](#_Toc515289764)

[5. 实验记录 7](#_Toc515289765)

[任务3 8](#_Toc515289766)

[1. 设计思想及存储单元分配 8](#_Toc515289767)

[2. 流程图 9](#_Toc515289768)

[3. 源代码 9](#_Toc515289769)

[4. 实验步骤 11](#_Toc515289770)

[5. 实验记录 11](#_Toc515289771)

[体会 11](#_Toc515289772)

[参考文献 11](#_Toc515289773)

# 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；
4. 熟悉跟踪与反跟踪的技术；
5. 提升对计算机系统的理解与分析能力。

# 实验内容

任务1：用三种方式获取中断类型码1H 、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，输入键盘上的小写字母时都变成了大写字母。

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命令行下键入小写字母，屏幕显示为大写，键入大写时不变。执行TD，在代码区输入指令“mov AX,0”，看是否都变成了大写。

（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（只需要还原中断矢量表的信息，先前驻留的程序可以不退出内存）。

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）在数据段定义一个待读取的CMOS内部单元的地址编号。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与操作系统显示的时间一致。

# 实验过程

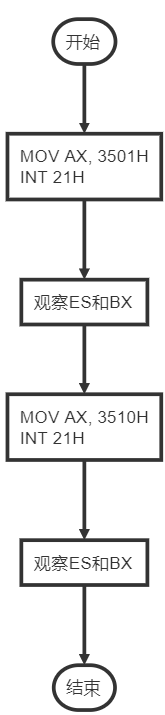
## 任务1

### 1. 设计思想及存储单元分配

使用3中方式读取中断类型码1H，10H对应的中断入口地址，由于题目已经给出了三种方法，在此就不再描述了。

### 2. 流程图

使用系统调用的流程图：



使用第三种方式的流程图：



### 3. 源代码

第二种方法的代码：

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

DATA ENDS

CODE SEGMENT USE16

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

START:

MOV AX, DATA

MOV DS, AX

; code after this

MOV AX, 3501H

INT 21H

MOV AX, 3510H

INT 21H

;stop here

MOV AH, 4CH

INT 21H

CODE ENDS

END START

第三种方法的代码：

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

; 获取中断物理地址的函数，参数AL

; 返回值AX, BX：中断的物理地址

GET\_INT PROC

XOR EBX, EBX

MOV BL, AL

XOR EAX, EAX

SHL BX, 2

PUSH DS

MOV AX, 0

MOV DS, AX

MOV AX, DS:[BX]

MOV BX, DS:[BX + 2]

POP DS

RET

GET\_INT ENDP

START:

; code after this

; 这个还行，直接读取

MOV AL, 1H

CALL GET\_INT

MOV AL, 10H

CALL GET\_INT

;stop here

MOV AH, 4CH

INT 21H

CODE ENDS

END START

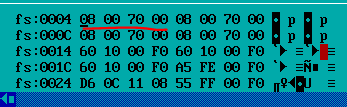
### 4. 实验步骤

（1）第一种方法直接进入td，在数据区寻找00:04H和00:40H即可找到1H中断和10H中断的地址。

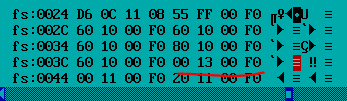
（2）第二种和第三种直接运行程序，查看返回值即可。

### 5. 实验记录

（1）第一种方式查看中断的方法，直接打开td，查看数据区，查看两个中断的地址如图：

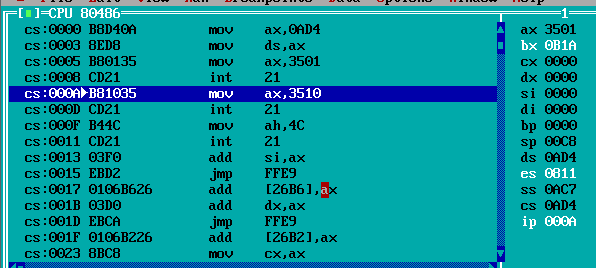


1H中断的处理的地址是0070H:0008H

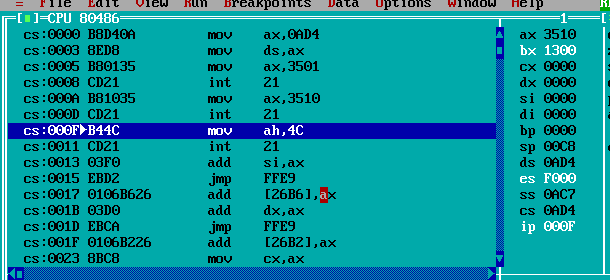


10H的中断处理地址是0F000H:1300H

（2）第二种方式结果的截图：

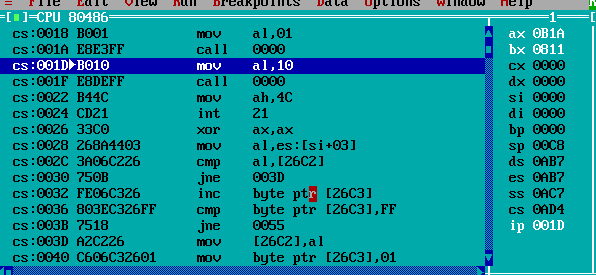


1H中断处理的地址是0811H:0B1AH

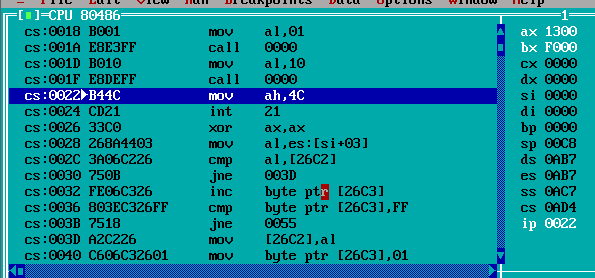


10H的中断处理地址是0F000:1300H

（3）第三种方式的结果：



1H的中断处理地址0811H:0B1AH



10H的中断处理地址是0F000H:1300H

## 任务2

### 1. 设计思想及存储单元分配

实验要求使用中断替换让程序能够将输入的小写字母转换为大写字母，其余不变。

设计思想：参考课本上的替换磁盘读取的那个程序，我的思想就是先保存旧的中断地址，使用自己的写的程序来替换掉旧中断，最后将代码驻留在内存中即可。

### 2. 流程图

### 3. 源代码

写中断的代码：

.386

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

OLD\_INT DW 0, 0

NEW16H PROC

PUSHF

CMP AH, 00H

JE GETKEY

CMP AH, 10H

JE GETKEY

; 调用旧中断

CALL DWORD PTR CS:OLD\_INT

JMP ENDINT

GETKEY:

CALL DWORD PTR OLD\_INT

; 获取了扫描码和ASCII码

; 判断大小写并转换

CMP AL, 'a'

JL ENDINT

CMP AL, 'z'

JG ENDINT

; 是小写字母

SUB AL, 32;变为大写

ENDINT:

IRET

NEW16H ENDP

START:

PUSH CS

POP DS

; code after this

MOV AX, 3516H

INT 21H

MOV OLD\_INT, BX

MOV OLD\_INT + 2, ES

; 替换中断

CLI;关闭中断响应

MOV DX, OFFSET NEW16H

MOV AX, 2516H

INT 21H

STI;开启中断

; 替换完毕，准备驻留内存

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

恢复中断的代码：

.386

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

START:

; 替换中断

CLI;关闭中断响应

MOV AX, 0F000H;

MOV DS, AX;

; 直接将旧中断写回

MOV DX, 11E0H

MOV AX, 2516H

INT 21H

STI;开启中断

MOV AH, 4CH

INT 21H

CODE ENDS

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

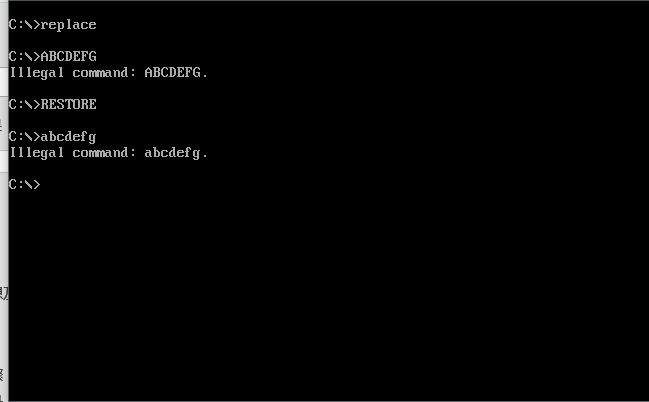
END START

### 4. 实验步骤

将程序编译连接得到可执行文件，直接执行文件即可将中断替换，想要恢复只需要执行恢复文件即可。

### 5. 实验记录

执行的记录如下：



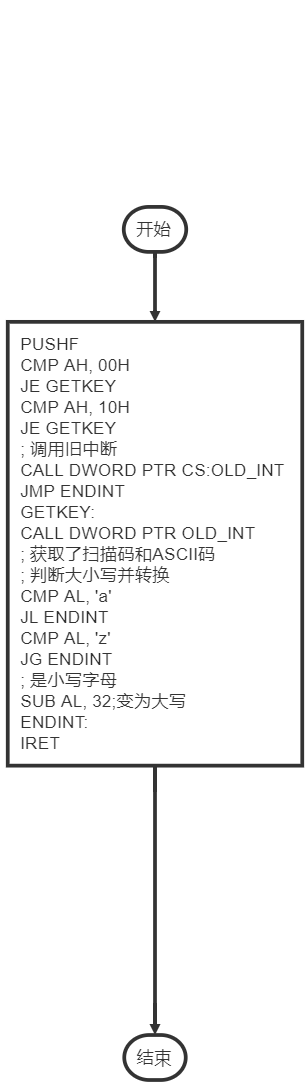
可见执行成功。

## 任务3

### 1. 设计思想及存储单元分配

实验要求从CMOS中读取数据并显示，我选择了最简单的显示时间，主要步骤就是向70H端口写入自己需要的数据号，再从71H中读取即可，其中需要注意的就是要注意只能读取一个byte的数据，包含了两个字符的BCD码，需要将它们分离开处理才能进行输出。

### 2. 流程图



### 3. 源代码

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

TIME DB '00:00:00$'

DATA ENDS

CODE SEGMENT USE16

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

PRINT\_TIME PROC

PUSH EAX

PUSH EDX

; hours

MOV AL, 4H

OUT 70H, AL

IN AL, 71H

MOV AH, AL

SHR AH, 4

AND AL, 0FH

ADD AH, '0'

ADD AL, '0'

MOV TIME[0], AH

MOV TIME[1], AL

; miuntes

MOV AL, 2

OUT 70H, AL

IN AL, 71H

MOV AH, AL

SHR AH, 4

AND AL, 0FH

ADD AH, '0'

ADD AL, '0'

MOV TIME[3], AH

MOV TIME[4], AL

; seconds

MOV AL, 0

OUT 70H, AL

IN AL, 71H

MOV AH, AL

SHR AH, 4

AND AL, 0FH

ADD AH, '0'

ADD AL, '0'

MOV TIME[6], AH

MOV TIME[7], AL

MOV DX, OFFSET TIME

MOV AH, 9

INT 21H

POP EDX

POP EAX

RET

PRINT\_TIME ENDP

START:

MOV AX, DATA

MOV DS, AX

; code after this

CALL PRINT\_TIME

;stop here

MOV AH, 4CH

INT 21H

CODE ENDS

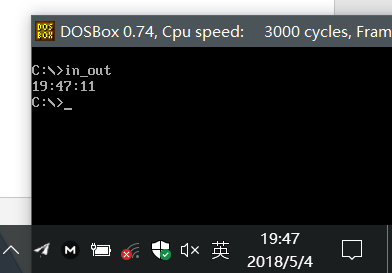
END START

### 4. 实验步骤

将代码编译连接得到可执行文件，执行程序就可以在屏幕上看到系统的时间。

### 5. 实验记录

执行程序如图：



对比可知获取的时间正确。

# 体会

本次实验让我理解了dos中的中断的具体物理地址的查找，如何替换已有中断，还有如何恢复中断，也学习了如何从CMOS中读入数据，对计算机的运行有了更深的理解。

# 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)