

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验五 中断程序设计**

**实验时间： 2017-11.14，18：30-21：50 实验地点： 南一楼804室6号实验台**

**指导教师： 许向阳**

**专业班级：信息安全 201601班**

**学 号： U201614804 姓 名： 黄永恒**

**同组学生： 无 报告日期： 2017年 11 月 15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](#_Toc500541156)

[2 实验内容 1](#_Toc500541157)

[3 实验过程 2](#_Toc500541158)

[3.1 任务一 2](#_Toc500541159)

[3.1.1 设计思想及存储单元分配 2](#_Toc500541160)

[3.1.2 源程序 2](#_Toc500541161)

[3.1.3 实验步骤 3](#_Toc500541162)

[3.1.4 实验记录与分析 3](#_Toc500541163)

[3.2 任务二 5](#_Toc500541164)

[3.2.1 设计思想及存储单元分配 5](#_Toc500541165)

[3.2.2 源程序 6](#_Toc500541166)

[3.2.3 实验步骤 9](#_Toc500541167)

[3.2.4 实验数据记录与分析 9](#_Toc500541168)

[3.3 选做题 10](#_Toc500541169)

[3.3.1 设计思想及存储单元分配 10](#_Toc500541170)

[3.3.2 源程序 10](#_Toc500541171)

[3.3.3 实验步骤 14](#_Toc500541172)

[3.3.4 实验数据记录及分析 14](#_Toc500541173)

[4 总结与体会 15](#_Toc500541174)

[参考文献 17](#_Toc500541175)

# 实验目的与要求

本次实验的主要目的与要求有下面4点，所有的任务都会围绕这4点进行，希望大家事后检查自己是否达到这些目的与要求。

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. 提升对计算机系统的理解与分析能力。

# 实验内容

**任务1. 用四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中断程序的入口地址；
3. 编写程序，直接读取相应内存单元，获取中断程序的入口地址；
4. 直接跟踪运行一个含有 INT 10H 语句的程序，按Alt+F7进入10H对应的中断处理程序,观察刚进入该程序时的地址。

使用TD观看有关信息即可，比较四种方式得到的中断程序的入口地址各是多少，它们是否相同。

**任务2. 编写一个接管时钟中断的中断服务程序并驻留内存，要求在屏幕的左上角显示时间。**

**要求**

1. 对程序中使用的DOS功能调用、BIOS调用都要予以说明。
2. 描述中断发生时，CPU的响应过程，以及中断处理程序运行到IRET时，CPU的处理过程。

**选做题.**

1. 编写一个程序，根据用户的输入，决定中断服务程序是否驻留内存。比较驻留与不驻留两种情况下，程序运行结束后的差别。
2. 编写一个中断服务程序的卸载程序，将时钟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，先前驻留的程序可以不仍保留在内存中）。

# 实验过程

## 任务一

### 设计思想及存储单元分配

1. 方法二：DOS系统功能调用方式

设计思想：利用DOS系统35号调用，取中断信息，最终将该中断的段地址值存储到ES中，偏移值存储到BX中。

存储单元分配：

DIZHI：字变量，存储类型码为10H的入口地址

寄存器分配：

AH：存储调用号

AL：存储中断号

1. 方法三：直接读取相应内存单元

设计思想：用一个寄存器直接取中断矢量表中类型码为10H的偏移地址，用另一个寄存器取段地址值。

存储单元分配

DIZHI2：字变量，存储类型码为10H的入口地址

寄存器分配

DX：存储中断矢量的偏移地址

SI：存储中断所在段地址值

### 源程序

.386

DATA SEGMENT USE16 PUBLIC 'DATA'

DIZHI DW ? , ?

DIZHI2 DW ? , ?

DATA ENDS

;--------------------------------------------------------------------------

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

;--------------------------------------------------------------------------

CODE SEGMENT USE16 PUBLIC 'CODE'

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

START: MOV AX, DATA

MOV DS, AX

MOV AX, 3510H ;方法二，DOS系统功能调用

INT 21H

MOV DIZHI, BX

MOV DIZHI+2, ES

MOV DX, FS: [40H] ;方法三，直接读取相应内存单元

MOV SI, FS: [42H]

MOV DIZHI2, DX

MOV DIZHI2+2, SI

INT 10H ;方法四，ALT+F7进入

MOV AH, 4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 准备上机环境，直接运行TD，观察内存中的中断矢量表
2. 准备上机环境，编辑、汇编、连接文件XSY。
3. 输入TD XSY，进入调试界面，单步运行到MOV AX,3510H处，运行后，观察BX和ES值。
4. 再运行到MOV DX,FS:[40H]处，运行后观察DX和SI值。
5. 运行到INT 10H处，按ALT+F7，进入对应的中断处理程序，观察刚进入时地址。

### 实验记录与分析

1. 实验环境条件：i7-5500U 3.0GHz，8G内存；WINDOWS 10下DOSBox0.73；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。
2. 直接运行TD，在数据区输入0:0，观察中断矢量表，如图3.1.2所示

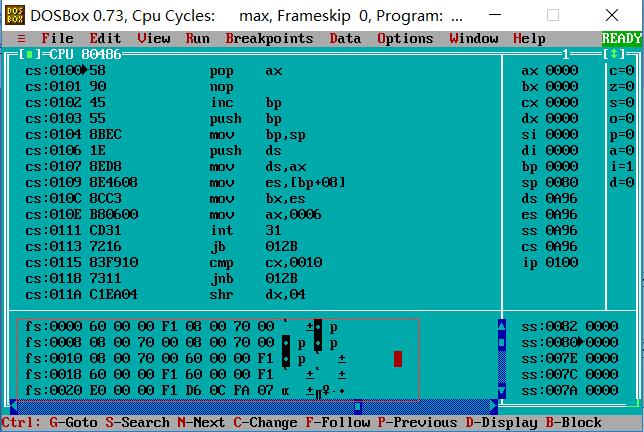


图3.1.2 中断矢量表

观察中断号为10H的地址值，如图3.1.3所示

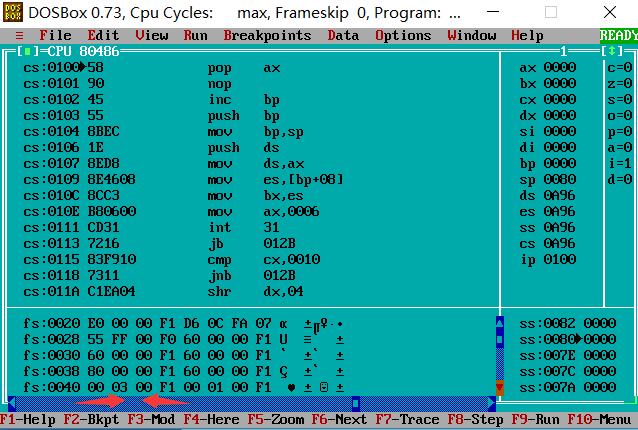


图3.1.3 中断号为10H的入口地址

1. 文件XSY.ASM编译连接正常，如图3.1.4所示

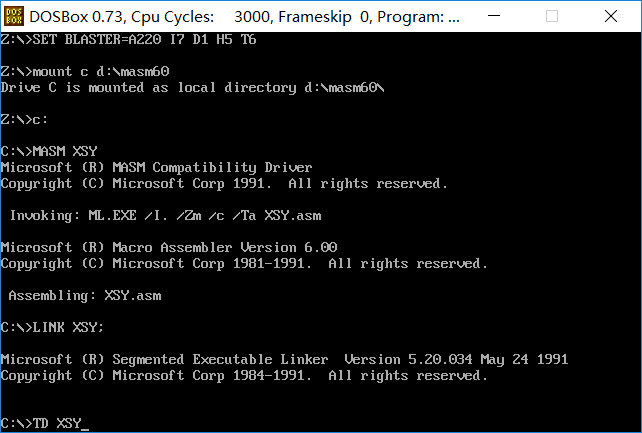


图3.1.4 正常编译连接

1. TD调试，运行到MOV AX,3510H处，运行后，观察BX和ES值，如图3.1.5所示

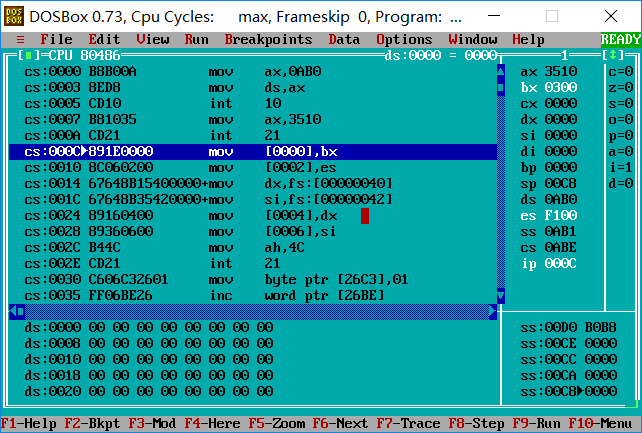


图3.1.5 DOS 系统功能调用后BX、ES值

1. 再运行到MOV DX,FS:[40H]处，运行后观察DX和SI值，如图3.1.6所示

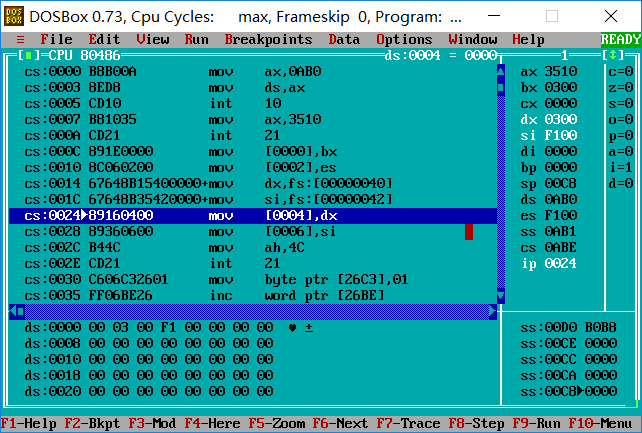


图3.1.6 直接访问内存单元，DX、SI存储入口地址

1. 运行到INT 10H处，按ALT+F7，进入对应的中断处理程序，观察刚进入时地址，如图3.1.7所示

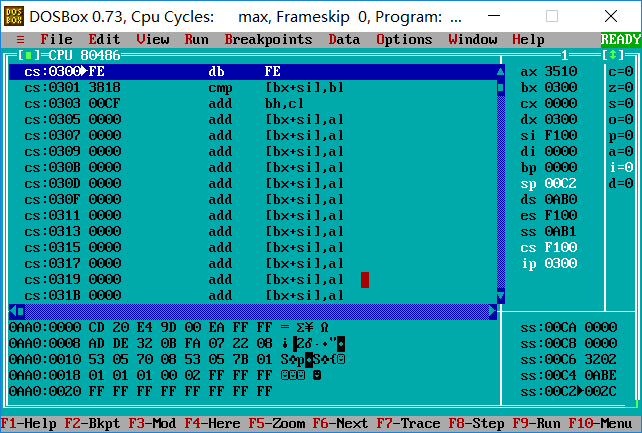


图3.1.7 ALT+F7进入中断处理程序后地址值

1. 综上比较四种方法得到的入口地址，段地址值均为 F100，偏移值均为0300。

## 任务二

### 设计思想及存储单元分配

1. 主程序

设计思想：通过DOS系统功能35号调用，获取原8号中断的中断处理程序的入口地址，并保存在相应的变量单元中，利用DOS系统功能25号调用，设置新的8号中断的中断处理程序的入口地址；然后判断是否有新的输入，如果有就退出程序。

存储单元分配：

MESSAGE：字节变量，作为字符串输出，提示

OLD\_INT：字变量，存储旧8号中断入口地址

寄存器分配：

AH：在DOS调用中存储编号

AL：在DOS调用中存储参数

1. 扩充8号中断处理程序

设计思想：在新的8号中断处理程序中可以先调用原8号中断处理程序的功能，对中断次数计数；采用倒计数的方法，当计数值由18减到0时，直接中断返回；当计数值减到0时，则重新给计数器赋初值，通过取时间子程序获取时间，利用BIOS‘INT 10H’功能显示信息；最后中断返回。

存储单元分配：

COUNT：字节变量，计数器

CURSOR：字变量，存储原光标地址

寄存器分配：

AH：在BIOS调用中存储编号

DH：存储光标位置行

DL：存储光标位置列

BH：在BIOS调用中作参数

CX：在BIOS调用中作参数

1. 取时间子程序

设计思想：通过I/O端口（70H/71H）直接读取RT/CMOS RAM中的时间信息

存储单元分配：

HOUR：字节变量，存储时

MIN：字节变量，存储分

SEC：字节变量，存储秒

### 源程序

.386

STACK SEGMENT USE16 STACK

DB 200 DUP (0)

STACK ENDS

CODE SEGMENT USE16

ASSUME CS: CODE, SS:STACK

COUNT DB 18

HOUR DB ? ,?, ':'

MIN DB ? , ?, ':'

SEC DB ? , ?

BUF\_LEN = $ - HOUR

CURSOR DW ?

OLD\_INT DW ?, ?

MESSAGE DB 0dh,0ah,'press any key to return',0dh,0ah,0dh,0ah,'$'

;--------------------------------------------------------------------------------------

; 扩充的 8号中断处理程序

NEW08H PROC FAR

PUSHF

CALL DWORD PTR OLD\_INT

DEC COUNT

JZ DISP

IRET

DISP: MOV COUNT,18

STI

PUSHA

PUSH DS

PUSH ES

MOV AX, CS

MOV DS, AX

MOV ES, AX

CALL GET\_TIME

MOV BH, 0

MOV AH, 3

INT 10H ; 读取光标位置 (DH,DL)=(行，列)

MOV CURSOR, DX

MOV DH, 0

MOV DL, 0

MOV BP, OFFSET HOUR

MOV BH, 0

MOV BL, 07H

MOV CX, BUF\_LEN

MOV AL, 1

MOV AH, 13H

INT 10H

MOV DX, CURSOR

ADD DL, BUF\_LEN ; 将 DL移到时间显示的结尾处

CMP DL, 80

JBE NEW08\_L1

INC DH

SUB DL, 81

MOV BH, 0

MOV AH, 3

INT 10H ; 读取光标位置 (DH,DL)=(行，列)

NEW08\_L1:

MOV BH, 0

MOV AH, 2

INT 10H

POP ES

POP DS

POPA

IRET

NEW08H ENDP

; -------------------------------GET\_TIME ------------------------------------------------

; 取时间

; 参考资料，CMOS数据的读写

GET\_TIME PROC

MOV AL, 4

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH, AL

AND AL,0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR HOUR, AX

MOV AL, 2

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH, AL

AND AL, 0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR MIN, AX

MOV AL, 0

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH, AL

AND AL, 0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR SEC, AX

RET

GET\_TIME ENDP

; -------------------------------------------------------------------------------------------------------

; 主程序开始

BEGIN:

; 显示定义在代码段中的 MESSAGE串中的内容，

PUSH CS

POP DS

; 获取原 8 号中断的中断处理程序的入口地址

MOV AX, 3508H

INT 21H!

MOV OLD\_INT, BX

MOV OLD\_INT+2, ES

; 设置新的 8号中断的中断处理程序的入口地址

MOV DX, OFFSET NEW08H

MOV AX, 2508H

INT 21H

LEA DX, MESSAGE

MOV AH, 9

INT 21H

NEXT: MOV AH, 0BH ; 判断有无击键，无则继续。有按键中止

INT 21H

CMP AL, 0

JZ NEXT

MOV DX, OFFSET BEGIN+15

MOV CL, 4

SHR DX, CL

ADD DX, 10H

MOV AL, 0

MOV AH, 31H

INT 21H

CODE ENDS

END BEGIN

### 实验步骤

* 1. 准备上机环境，编辑、汇编、连接文件XSY4.ASM。
  2. 运行程序，观察是否有中断，界面左上角是否有时间显示
  3. 按任意键退出，观察是否驻留内存（继续显示）

### 实验数据记录与分析

1. 实验环境条件：i7-5500U 3.0GHz，8G内存；WINDOWS 10下DOSBox0.73；EDIT.EXE 2.0；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。
2. 运行该程序，如图3.2.1所示

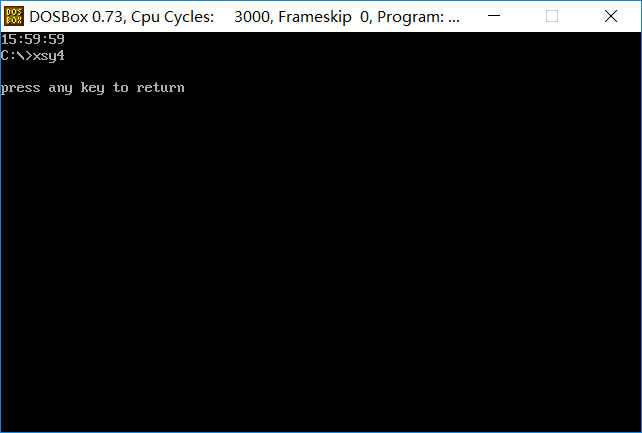


图3.2.1 运行程序后，每隔一秒新8号中断显示一次时间

1. 按任意键退出，如图3.2.2所示

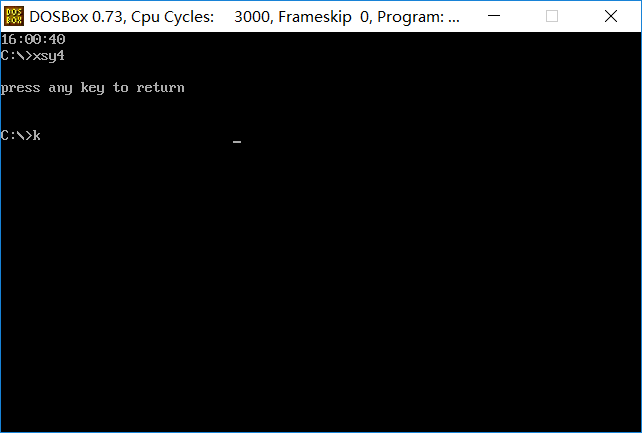


图3.2.2 退出程序后，中断服务程序仍驻留内存，继续显示

1. 程序中使用的DOS功能调用有：INT 21H

09H号调用，入口参数：DS：[DX] = 缓冲区首址，用于字符串输出；

35H号调用，入口参数：（AL）= 中断类型号，出口参数：ES：[BX] = 入口地址，用于取中断信息；

25H号调用，入口参数：（AL）= 中断类型号，DS：[DX] = 入口地址，用于置中断矢量；

0BH号调用，出口参数：（AL）= 00无键入，（AL）= 11有键入，用于判断是否有输入；

31H号调用，入口参数：（AL）= 退出码，（DX）= 驻留节数，用于终止用户程序并驻留在内存。

1. 程序中使用的BIOS 功能调用有：INT 10H

03H号调用，入口参数：（BH）= 页号，出口参数：（DH，DL）= 行，列

（CH，CL）= 当前光标模块，用于读取光标位置；

13H号调用，入口参数：（AL）= 1，写后修改光标位置，（BH）= 页号，（CX）= 字符串长度，（DH，DL）= 行，列，ES：[BP] = 待显示字符串的起始地址，用于写字符串到指定的光标位置；

02H号调用，入口参数：（DH，DL）= 行，列 ，（BH）= 页号，用于设置光标位置。

1. 中断发生时，CPU保护断点，即保存下一将要执行的指令的地址，将标志寄存器、CS、IP依次压入堆栈中，然后根据中断类型查找入口地址，进入相应的中断处理程序，然后执行中断处理程序。

在中断处理程序中运行到IRET时，将之前压入栈中的IP、CS、标志寄存器依次出栈，退出中断处理程序，返回到主程序，CPU继续执行。

## 选做题

### 设计思想及存储单元分配

设计思想：在任务二的基础上，可以考虑在主程序内加入输入字符语句，然后根据输入的字符进行判断是否需要驻留，在取时间子程序中可以将年月日也取出。

卸载程序可以考虑将原来的入口地址再存到中断矢量表中。

存储单元分配（新增）：

PROMPT：字节变量，字符串输出提示信息

CHOOSE：字变量，存储输入的字符

### 源程序

（一）

.386

STACK SEGMENT USE16 STACK

DB 200 DUP (0)

STACK ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

OLD\_INT DW ?, ?

COUNT DB 18

YEAR DB '2', '0', ?, ?,'/'

MONTH DB ?, ?, '/'

DATE DB ?, ?, ' '

HOUR DB ?, ?, ':'

MIN DB ?, ?, ':'

SEC DB ?, ?

BUF\_LEN = $ - YEAR

CURSOR DW ?

MESSAGE DB 0dh,0ah,'press any key to return',0dh,0ah,0dh,0ah,'$'

PROMPT DB 0DH, 0AH, ' Please select: '

DB 0DH, 0AH, ' 1: new 8 interrupt, and restore when return'

DB 0DH, 0AH, ' 2: new 8 interrupt, and resident when return'

DB 0DH, 0AH, '$'

CHOOSE DB ?

;--------------------------------------------------------------------------------------

; 扩充的 8号中断处理程序

NEW08H PROC FAR

PUSHF

CALL DWORD PTR OLD\_INT

DEC COUNT

JZ DISP

IRET

DISP: MOV COUNT,18

STI

PUSHA

PUSH DS

PUSH ES

MOV AX, CS

MOV DS, AX

MOV ES, AX

CALL GET\_TIME

MOV BH, 0

MOV AH, 3

INT 10H ; 读取光标位置 (DH,DL)=(行，列)

MOV BP, OFFSET YEAR

MOV BH, 0

MOV BL, 07H

MOV CX, BUF\_LEN

MOV AL, 1

MOV AH, 13H

INT 10H

MOV BH, 0

MOV AH, 3

INT 10H ; 读取光标位置 (DH,DL)=(行，列)

NEW08\_L1:

MOV BH, 0

MOV AH, 2

INT 10H

POP ES

POP DS

POPA

IRET

NEW08H ENDP

; -------------------------------GET\_TIME ------------------------------------------------

; 取时间

; 参考资料，CMOS数据的读写

GET\_TIME PROC

MOV AL, 9

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH,AL

AND AL,0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR YEAR+2, AX

MOV AL, 8

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH,AL

AND AL,0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR MONTH, AX

MOV AL, 7

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH,AL

AND AL,0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR DATE, AX

MOV AL, 4

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH,AL

AND AL,0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR HOUR, AX

MOV AL, 2

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH, AL

AND AL, 0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR MIN, AX

MOV AL, 0

OUT 70H, AL

JMP $+2

IN AL, 71H

MOV AH, AL

AND AL, 0FH

SHR AH, 4

ADD AX, 3030H

XCHG AH, AL

MOV WORD PTR SEC, AX

RET

GET\_TIME ENDP

; -------------------------------------------------------------------------------------------------------

; 主程序开始

BEGIN:

; 显示定义在代码段中的 MESSAGE串和PROMPT串中的内容，

PUSH CS

POP DS

LEA DX, PROMPT

MOV AH, 9

INT 21H

MOV AH,1

INT 21H

MOV CHOOSE,AL

LEA DX, MESSAGE

MOV AH, 9

INT 21H

; 获取原 8 号中断的中断处理程序的入口地址

MOV AX, 3508H

INT 21H

MOV OLD\_INT, BX

MOV OLD\_INT+2, ES

; 设置新的 8号中断的中断处理程序的入口地址

MOV DX, OFFSET NEW08H

MOV AX, 2508H

INT 21H

NEXT: MOV AH, 0BH ; 判断有无击键，无则继续。有按键中止

INT 21H

CMP AL, 0

JZ NEXT

CMP CHOOSE,'1'

JZ RESTORE

MOV DX, OFFSET BEGIN+15

MOV CL, 4

SHR DX, CL

ADD DX, 100H

MOV AL, 0

MOV AH, 31H

INT 21H

; 原来的程序是将中断处理程序的入口地址复原

RESTORE:LDS DX, DWORD PTR OLD\_INT

MOV AX, 2508H

INT 21H

MOV AH, 4CH

INT 21H！

CODE ENDS

END BEGIN

（二）

.386

STACK SEGMENT USE16 STACK

DB 200 DUP (0)

STACK ENDS

DATA SEGMENT USE16

OLD\_INT DW ?, ?

DATA ENDS

CODE SEGMENT USE16

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

START: MOV AX, DATA

MOV DS, AX

MOV AX, 3508H

INT 21H

MOV AX, ES:[0H]

MOV OLD\_INT, AX

MOV AX, ES:[2H]

MOV OLD\_INT+2, AX

LDS DX, DWORD PTR OLD\_INT

MOV AX, 2508H

INT 21H

MOV AH, 4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 准备上机环境，编辑、汇编、连接文件FJT.ASM、XZ.ASM。
2. 运行程序，首先选择不驻留，按任意键退出，观察退出后程序运行情况。
3. 再次运行程序，选择驻留，按任意键退出，观察退出后程序运行情况。
4. 运行XZ程序，观察显示器显示情况。

### 实验数据记录及分析

1. 选择不驻留，按任意键退出，如图3.3.1所示

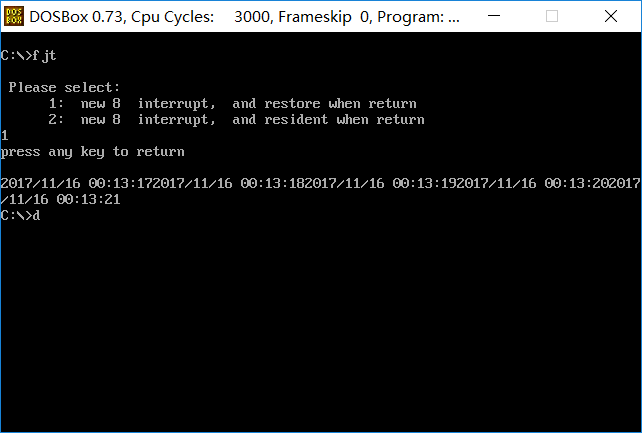


图3.3.1 不驻留退出程序

1. 选择驻留，按任意键退出，如图3.3.2所示

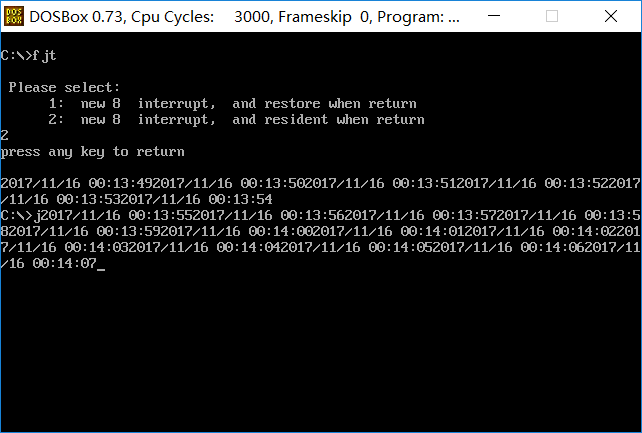


图3.3.2 驻留退出程序

1. 运行XZ程序，如图3.3.3所示

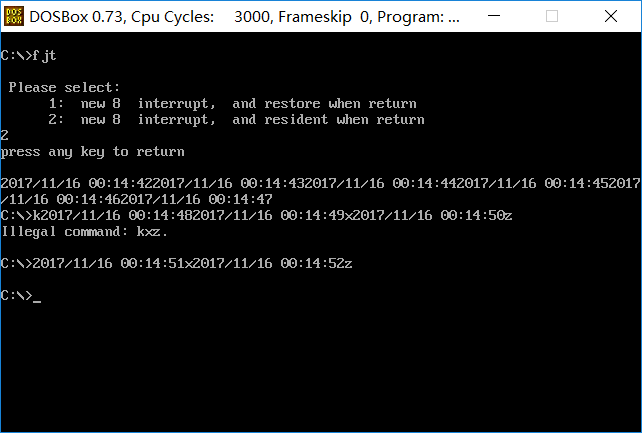


图3.3.3 卸载程序后，不再继续显示

# 总结与体会

本次实验主要是围绕中断这一知识点展开的。

在任务一中，我了解了中断矢量表，并且通过自己尝试，编写程序获取了中断类型为10H的处理程序入口地址，并且进入该处理程序，观察中断是如何进行的。

在任务二中，我学会了如何接管某一中断处理程序以扩展其功能，并且了解了CMOS相关知识，通过I/O访问获取时间，并且学会了如何将中断程序驻留内存。

在附加题中，也是应用中断程序驻留内存相关知识，以及复原中断处理程序、

由于老师给了我们参考代码并且书上也有相关代码，因此本次实验实际撰写的代码量很小，因此我认为这次实验主要目的是掌握中断是如何发生的，了解CPU在中断发生时是如何处理的，借此提升对计算机系统的理解与分析能力。

通过每次实验，我发现自己的这个能力在渐渐增长，实践出真知！

# 参考文献

[1] 王元珍 曹忠升 韩宗芬.《80x86汇编语言程序设计》“第六章 输入输出和WIN32编程”.华中科技大学出版社