

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称： 实验四 中断与反跟踪**

**实验时间： 2017-4-19，14：00-17：30 实验地点： 南一楼803室99号实验台**

**指导教师： 张勇**

**专业班级： 计算机科学与技术201307班**

**学 号： U201314969 姓 名： 王镇宇**

**同组学生： 无 报告日期： 2017年 4 月 27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 2](#_Toc480376032)

[2 实验内容 2](#_Toc480376033)

[3 实验过程 4](#_Toc480376034)

[3.1 任务1 4](#_Toc480376035)

[3.1.1 设计思想及存储单元分配 4](#_Toc480376036)

[3.1.2 流程图 4](#_Toc480376037)

[3.1.3 源程序 4](#_Toc480376038)

[3.1.4 实验步骤 5](#_Toc480376039)

[3.1.5 实验记录 5](#_Toc480376040)

[3.2 任务2 5](#_Toc480376041)

[3.2.1 设计思想及存储单元分配 5](#_Toc480376042)

[3.2.2 流程图 6](#_Toc480376043)

[3.2.3 源程序 6](#_Toc480376044)

[3.2.4 实验步骤 8](#_Toc480376045)

[3.2.5 实验记录 8](#_Toc480376046)

[3.3 任务3 9](#_Toc480376047)

[3.3.1 设计思想及存储单元分配 9](#_Toc480376048)

[3.3.2 流程图 9](#_Toc480376049)

[3.3.3 源程序 9](#_Toc480376050)

[3.3.4 实验步骤 11](#_Toc480376051)

[3.3.5 实验记录 11](#_Toc480376052)

[3.4 任务4 11](#_Toc480376053)

[3.4.1 设计思想及存储单元分配 11](#_Toc480376054)

[3.4.2 流程图 12](#_Toc480376055)

[3.4.3 源程序 12](#_Toc480376056)

[3.4.4 实验步骤 18](#_Toc480376057)

[3.4.5 实验记录 18](#_Toc480376058)

[3.5 任务5 18](#_Toc480376059)

[3.5.1 设计思想及存储单元分配 18](#_Toc480376060)

[3.5.2 流程图 18](#_Toc480376061)

[3.5.3 源程序 18](#_Toc480376062)

[3.5.4 实验步骤 19](#_Toc480376063)

[4 体会 19](#_Toc480376064)

[参考文献 20](#_Toc480376065)

# 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

任务4：数据加密与反跟踪

在实验一的学生成绩查询程序的基础上，增加查询前输入密码的功能，密码不对则程序退出，只有密码正确之后才能完成后续的功能。密码采用密文的方式存放在数据段中。各科成绩也以密文方式存放在数据段中。加密方法自选。

可以采用计时、中断矢量表检查、堆栈检查、间接寻址等方式中的一种或多种方式反跟踪（建议只采用一到两种反跟踪方法，重点是深入理解和运用好所选择的反跟踪方法）。

成绩表中要有编程者自己的名字（姓的全拼+名字的拼音首字母，但大小写可以随意组合）和各科成绩（姓名和成绩都密文存放）。成绩表中只需要定义三个学生的信息即可。

提示：为了使源程序的数据段中定义的密码、学生姓名、各科成绩能在汇编之后变成密文（也就是在最后交付出去的执行程序中看不到明文），可以使用数值运算符（参见教材P48）对变量的初始值进行变换。例如，如果想使语文成绩90分变成密文，加密算法是与密钥字符“W”做异或运算，则可写成：

YUWEN DB 90 XOR ‘W’

任务5：跟踪与数据解密

解密同组同学的加密程序，获取该同学的各科成绩。

注意：两人一组，每人实现一套自己选择的加密与反跟踪方法，把执行程序交给对方解密（解密时间超过半小时的，说明反跟踪方法基本有效）。如何设计反跟踪程序以及如何跟踪破解，是本次实验报告中重点需要突出的内容。（分组可以按照学号顺序依次构成两人一组。也可以自行调整。如果班上人数是奇数，则三人一组，甲解密乙的，乙解密丙的，丙解密甲的）

# 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；

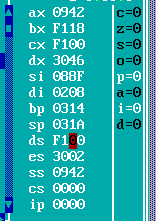
3.将程序renwu62编译、链接，调试直到没有错误。运行td renwu62，在td中单步调试，观察寄存器的值。

### 实验记录

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。

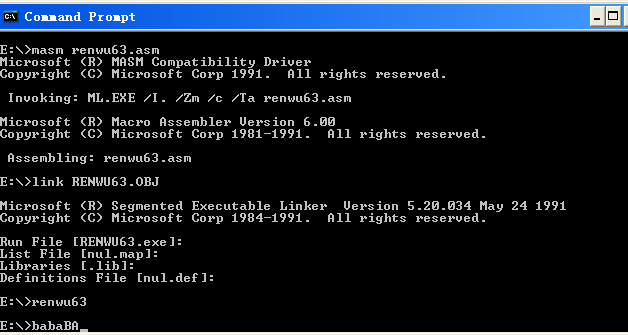
3. 执行该程序输入a/A,观察是否输出b/B;输入b/B,观察是否输出a/A。

### 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。



## 任务4

### 设计思想及存储单元分配

此次实验主要是两部分内容：1.实现正常功能模块（即实验2所做的内容）2.加密模块：要求提供验证密码功能（加密算法设计：使用的是简单的算数逻辑运算，具体是将姓名与s异或，成绩与Bat异或，密码为Bat，采用函数(X-29H)\*3对保存的密码进行编码；反跟踪功能设计：在各个模块中穿插反跟踪代码，防止破解人员利用td破解）。

### 流程图



### 源程序

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

;

DATA SEGMENT USE16

NUM DD 3

RADX DD 10

BASE DB 10

D1 DB 0DH,0AH,'STUDENT NAME:$'

D2 DB 0DH,0AH,'GRADE:$'

PROMPT db 0dh, 0ah, 'please enter password: $'

MSG db 0dh, 0ah, 'please input target name : $'

BUF DB 'z' XOR 's','h'XOR 's','a'XOR 's', 7 DUP(0)

DB 100 XOR 'B' ,85 XOR 'a', 80 XOR 't', ?

DB 'l' XOR 's','i'XOR 's','s'XOR 's','i' XOR 's',6 DUP(0)

DB 80 XOR 'B', 98 XOR 'a', 70 XOR 't',?

PWD DB 3 XOR 'C' ;密码串的长度为3，采用与常数43H异或的方式编码成密文

DB ('B' -29H)\*3 ;真实密码为Bat。采用函数(X-29H)\*3对保存的密码进行编码。

DB ('a' -29H)\*3

DB ('t' -29H)\*3

DB 0A1H,5FH,0D3H ;用随机数填充密码区到6个字符，防止破解者猜到密码长度

;

IN\_PWD DB 7 ;输入密码，最大长度6个字符

DB ?

DB 7 DUP(0)

;

INPUT DB 11

DB ?

DB 11 DUP(0)

P1 DW PASS1 ;地址表

E1 DW OVER

P2 DW PASS2

P3 DW PASS3

DATA ENDS

;

CODE SEGMENT USE16

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

START: MOV AX,DATA

MOV DS,AX

LEA DX,PROMPT

MOV AH,9

INT 21H

LEA DX,IN\_PWD ;输入密码字符串

MOV AH,10

INT 21H

cli ;计时反跟踪开始

mov ah,2ch

int 21h

push dx ;保存获取的秒和百分秒

MOV CL,IN\_PWD+1 ;比较输入的串长与密码长度是否一样

XOR CL,'C'

SUB CL,PWD

MOVSX BX,CL

ADD BX,OFFSET P1

mov ah,2ch ;获取第二次秒与百分秒

int 21h

sti

cmp dx,[esp] ;计时是否相同

pop dx

jz OK1 ;如果计时相同，通过本次计时反跟踪

mov bx,offset E1 ;如果计时不同，则把转移地址偏离P1

OK1: mov bx,[bx]

cmp word ptr cs:[bx],0B60FH ;是否是PASS1处的指令，其实是用于判断前面比较的

;串长是否相同

jz OK2

jmp E1

OK2: jmp bx

db 'How to go' ;定义的冗余信息

PASS1: MOVZX CX,IN\_PWD+1

cli ;堆栈检查反跟踪

push P2 ;PASS2的地址压栈

MOV SI,0

MOV DL,3

pop ax

mov bx,[esp-2] ;把栈顶上面的字（PASS2的地址）取到

sti

jmp bx ;如果被跟踪，将不会转移到PASS2

db 'i donot know！'

PASS2: MOVZX AX,IN\_PWD+2[SI] ;比较密码是否相同。把输入的串变成密文，与保存的密文比较

SUB AX,29H

MUL DL

CMP AL,PWD+1[SI]

JNZ ERR2

INC SI

DEC CX

CMP CX,0

JNE PASS2

JMP PASS3

ERR2: MOV EBX,OFFSET P1

MOV EDX,1

JMP WORD PTR [EBX+EDX\*2] ;指向OVER

db 'YES,get it'

;

PASS3: ;　正常功能区

lea dx, msg ; 输出提示信息 - 请输入姓名

mov ah, 9h

int 21h

lea dx, input ; 读入学生姓名, 以 '$' 符号结尾

mov ah, 0ah

int 21h

lea esi, input+2

lea edi, buf

xor ebx, ebx

xor ecx, ecx

search\_loop:

cmp ecx, num

jge search\_finish

xor eax, eax

search\_inner:

cmp eax, radx

jge search\_finish

mov dl, [esi + eax]

xor dl, 's'

cmp [edi + ebx], dl

jnz search\_next

cmp byte ptr [edi + ebx + 1], 0

jz search\_finish

inc eax

inc ebx

jmp search\_inner

db 'the future will be better tomorrow'

search\_next:

inc ecx

imul ebx, ecx, 14

jmp search\_loop

db 'welcome to hust'

search\_finish:

xor ebx, ebx

output:

sub ebx,ecx

imul bx,14 ; 根据目标学生下标值, 找到分数缓冲区首地址

mov si,0

mov ax, 0

mov dx, 0

mov al,buf+10[bx+si]

xor al,in\_pwd+2[si]

add ax, ax

inc si

mov dl, buf+10[bx+si]

xor dl,in\_pwd+2[si]

add ax, dx ; al = zh \* 2 + ma

inc si

mov dl, buf+10[bx+si]

xor dl,in\_pwd+2[si]

sar dl, 1

add ax, dx ; al = zh \* 2 + ma + en / 2

sal ax, 1 ; al = 2 \* al

mov dx, 7

idiv dl ; al = al / 7

inc si

mov buf+10[bx+si], al ; avg = al ( al / 3.5)

push ax

mov dl, 0ah

mov ah, 2h

int 21h

mov dl, 0dh

mov ah, 2h

int 21h

pop ax

cmp al, 90 ; switch 语句

jge gradea

cmp al, 80

jge gradeb

cmp al, 70

jge gradec

cmp al, 60

jge graded

jmp gradee

db 'my sunshine'

gradea: mov dl, 41h

mov ah, 2h

int 21h

jmp pass3

gradeb: mov dl, 42h

mov ah, 2h

int 21h

jmp pass3

gradec: mov dl, 43h

mov ah, 2h

int 21h

jmp pass3

graded: mov dl, 44h

mov ah, 2h

int 21h

jmp pass3

gradee: mov dl, 45h

mov ah, 2h

int 21h

jmp pass3

OVER:

MOV AH,4CH

INT 21H

CODE ENDS

END START

### 实验步骤

1. 绘制程序流程图；

2. 根据程序流程图，编写shiyan7.asm；

3. 编译、链接源程序，生成可执行目标文件shiyan7.exe;

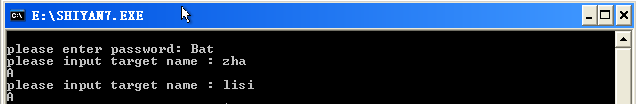
4. 运行shiyan7.exe，检查程序具有正常功能;

5. 输入正确密码后，键入学生姓名，可以正常查询出成绩；

6. 输入错误密码后，程序直接终止运行；。

### 实验记录

1. 实验环境条件：P3 1GHz，256M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. 输入密码Bat后程序运行如下



当输入错误的密码时程序直接停止运行，当破解人员在td里调试时，由于执行两条指令的时间远大于执行程序的时长，通过检测这一时差，若两次计时调用间时长过长，直接结束程序，阻止了破解人员进一步破解；当破解人员利用反汇编工具运行程序时，栈顶数据的值被修改，可通过此检测程序是否是在反汇编环境下执行；若程序在异常环境下运行，则立即终止程序。

## 任务5

### 设计思想及存储单元分配

在td里单步执行待破解的程序，可以很直观地获得程序的执行流程和完成的操作。若碰到反跟踪程序段，配合设置断点的方法，绕过反跟踪程序段设置的陷阱。

### 流程图

无

### 源程序

无

### 实验步骤

1. 运行TD，开始破解密码；

2. 当遇到疑似反跟踪代码时，若其中含有功能代码，则在其尾部设置断点，直接运行过去；

3. 当遇到疑似反跟踪代码时，若其中不含功能代码，纯粹为反跟踪代码，则直接修改IP寄存器的值，跳过此段代码的执行;

4. 在验证密码与计算成绩处，找出加密手段；在相关数据段，找出暗文数据;

5. 利用暴力破解法,编写破解脚本，结合加密手段与暗文，得到明文数据;

6. 记录明文数据，并利用源代码进行验证，查看是否成功破密;

# 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。初步熟悉了一些跟踪与反跟踪手段：中断向量检测、计时检测、堆栈检测。提升了自己对计算机系统的理解与分析能力，懂得如何进行简单的数据加密。这次实验收获很大，相信对今后的编程有很大的帮助。

# 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)