

**课 程 实 验 报 告**

**课程名称： 汇编语言程序设计实验**

**实验名称：** 实验七 跟踪与反跟踪程序设计

**实验时间： 2016-5-3 ，14：30-17：30 实验地点： 南一楼804室102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： U2014414808 姓 名： 王林**

**同组学生： 无 报告日期： 2016年 5 月 3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： 王林

日期： 2016.5.3

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](#_Toc450380225)

[实验过程 1](#_Toc450380226)

[7.1任务1 1](#_Toc450380227)

[7.1.1设计思想及存储单元分配 1](#_Toc450380228)

[7.1.2流程图 2](#_Toc450380229)

[7.1.3源程序 2](#_Toc450380230)

[7.1.4实验步骤 9](#_Toc450380231)

[7.1.5实验记录 9](#_Toc450380232)

[7.2任务2 9](#_Toc450380233)

[7.2.1设计思想及存储单元分配 10](#_Toc450380234)

[7.2.2流程图 10](#_Toc450380235)

[7.2.3源程序 10](#_Toc450380236)

[7.2.4实验步骤 10](#_Toc450380237)

[7.2.5 实验记录 10](#_Toc450380238)

[体会 12](#_Toc450380239)

[参考文献 13](#_Toc450380240)

实验目的与要求

（1） 熟悉跟踪与反跟踪的技术；

（2） 提升对计算机系统的理解与分析能力。

**实验内容**

任务1：数据加密与反跟踪

在实验二的基础上，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。

成绩表中要有自己的名字和各科成绩（密文存放）

**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，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2：跟踪与数据解密

解密同组同学的加密程序，获取该同学的成绩。

实验过程

7.1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。

7.1.1设计思想及存储单元分配

首先在数据区设置密码，并且对密码和密码长度进行加密。然后对数据区其他数据进行加密，包括姓名和各科成绩。

进入系统前提示输入密码，输入正确密码才能进入系统。匹配密码可以通过明文匹配和暗文匹配两种方式。

进入系统后，提示输入学生姓名来进行学生信息查找。匹配姓名字符串可以通过明文匹配和暗文匹配两种方式。试验中采用暗文匹配。只要输入的名字加密后能够暗文匹配就行。

匹配到学生姓名后，就计算学生成绩并黑色素出等级。则先对加密后的成绩解密，然后计算。

同时要设置反跟踪，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。本次实验中，我通过增加冗余代码方式来防止静态反汇编，设置计时来反动态跟踪。

7.1.2流程图



**图 7-1 任务一流程图**

7.1.3源程序

【lab-7.asm】

;lab-2

;source code

;@author Strawberrylin

.386

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

DATA SEGMENT USE16

BUF DB 15

DB ?

DB 15 DUP(0)

;对密码和密码长度加密

PASWORD DB 7 XOR 'C','W' XOR 'W','a' XOR 'W','n' XOR 'W',

'g' XOR 'W','l' XOR 'W','i' XOR 'W','n' XOR 'W'

STU DB 'W' XOR 'W','A' XOR 'W','N' XOR 'W','G' XOR 'W',

'L' XOR 'W','I' XOR 'W','N' XOR 'W',0 XOR 'W',

2 DUP(0),100 XOR 'W',90 XOR 'W',90 XOR 'W',0

DB 'Z' XOR 'W','H' XOR 'W','A' XOR 'W','N' XOR 'W',

'G' XOR 'W','S' XOR 'W','A' XOR 'W','N' XOR 'W',

0 XOR 'W',0,80 XOR 'W',70 XOR 'W',60 XOR 'W',0

DB 'L' XOR 'W','I' XOR 'W','S' XOR 'W','I' XOR 'W',0 XOR 'W',

5 DUP(0),0 XOR 'W',10 XOR 'W',40 XOR 'W',0

BUF1 DB 'Input Student Name:',0AH,0DH,'$'

BUF2 DB 'No This Student...',0AH,0DH,'$'

BUF3 DB 'ENTER Y TO EXIT OR N GO ON:',0AH,0DH,'$'

BUF4 DB 'Input the password:',0AH,0DH,'$'

DATA ENDS

CODE SEGMENT USE16

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

START:

MOV AX, DATA

MOV DS, AX

LEA DX, BUF4

MOV AH, 9 ;9TH FUNCTION CALLS

INT 21H

LEA DX, BUF ;INPUT STUDENT NAME

MOV AH, 10 ;10TH FUNCTION CALLS

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

LEA DI, PASWORD

LEA SI, BUF+2

MOV AL, BUF+1

MOV AH, 0

CMPPASWL:

MOV CL, PASWORD

XOR CL, 'C'

MOV CH, 0

CMP AX, CX

JNZ EXIT

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 CMPPASWN

JMP EXIT

MOV DL, 0AH ;冗余代码

MOV AH, 2

INT 21H

MOV DL, 0DH

MOV AH, 2

INT 21H

CMPPASWN:

MOV CL, PASWORD

XOR CL, 'C'

MOV CH, 0

MOV DX, CX

JMP CMPPASW

MOV DL, 0AH ;冗余代码

MOV AH, 2

INT 21H

MOV DL, 0DH

MOV AH, 2

INT 21H

CMPPASW:

MOV AL, [SI]

MOV BL, 1[DI]

XOR BL, 'W'

CMP AL, BL

JNZ EXIT

DEC DX

JZ INSYS

INC DI

INC SI

JMP CMPPASW

MOV DL, 0AH ;冗余代码

MOV AH, 2

INT 21H

MOV DL, 0DH

MOV AH, 2

INT 21H

INSYS:

MOV CX, 3H

LEA DX, BUF1

MOV AH, 9 ;9TH FUNCTION CALLS

INT 21H

LEA DX, BUF ;INPUT STUDENT NAME

MOV AH, 10 ;10TH FUNCTION CALLS

INT 21H

LEA DI, STU

MOV DL, 0AH

MOV AH, 2

INT 21H

MOV DL, 0DH

MOV AH, 2

INT 21H

LOAPI:

LEA SI, BUF+2

MOV AL, BUF+1

MOV AH, 0

MOV BX, AX

MOV AL, 0

MOV [SI+BX], AH

MOV BX, 0

LOAPJ:

MOV AL, [BX+SI]

MOV DL, [BX+DI]

XOR DL, 'W'

CMP AL, DL

JNZ NEXT

CMP DL, 0

JZ FIGURE

INC BX

JMP LOAPJ

NEXT:

DEC CX

JZ REINPUT

ADD DI, 14

JMP LOAPI

FIGURE:

MOV AL, [DI+10]

MOV AH, 0

XOR AL, 'W'

ADD AX, AX

ADD AX, AX

MOV DL, [DI+11]

MOV DH, 0

XOR DL, 'W'

ADD AX, DX

ADD AX, DX

MOV DL, [DI+12]

MOV DH, 0

XOR DL, 'W'

ADD AX, DX

MOV BL, 7

DIV BL

MOV [DI+13], AL

JMP COMPARE

COMPARE:

MOV AL, [DI+13]

CMP AL, 90

JGE A

CMP AL, 80

JGE B

CMP AL, 70

JGE E

CMP AL, 60

JGE D

CMP AL, 50

JL F

A:

MOV DL, 41H

MOV AH, 2

INT 21H

MOV DL, 0AH

INT 21H

JMP EXIT

B:

MOV DL, 42H

MOV AH, 2

INT 21H

MOV DL, 0AH

INT 21H

JMP EXIT

E:

MOV DL, 43H

MOV AH, 2

INT 21H

MOV DL, 0AH

INT 21H

JMP EXIT

D:

MOV DL, 44H

MOV AH, 2

INT 21H

MOV DL, 0AH

INT 21H

JMP EXIT

F:

MOV DL, 46H

MOV AH, 2

INT 21H

MOV DL, 0AH

INT 21H

JMP EXIT

REINPUT:

LEA DX, BUF2

MOV AH, 9

INT 21H

JMP START

EXIT:

LEA DX, BUF3

MOV AH, 9 ;9TH FUNCTION CALLS

INT 21H

LEA DX, BUF ;INPUT STUDENT NAME

MOV AH, 10

int 21H

LEA SI, BUF+2

MOV AL, [SI]

CMP AL, 'Y'

JNZ START

MOV AH, 4CH

INT 21H

CODE ENDS

END START

7.1.4实验步骤

1、按要求修改实验2的源程序，在数据段加入密码，并且对数据段进行加密。增加冗余代码并且设置计时来反跟踪。

2、编译、链接、生成可执行程序，并且测试通过，没有错误。

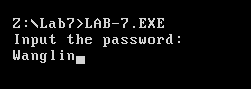
3运行exe程序，输入自己设置的密码，检测密码是否正确；进入系统后，输入自己姓名，检测是否可以输出正确的成绩等级。

7.1.5实验记录

1、实验环境条件：VM XP虚拟机，处理器: 2，内存：1GB；命令行窗口；Sublime Text 2；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2、对实验2的源程序修改，检测是否通过。

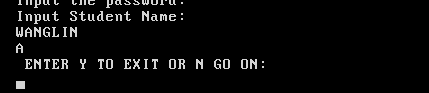
1. 检测编译是否通过；
2. 依照屏幕提示输入密码，检测密码是否正确；



**图 7-2 测试验证**

验证密码正确

检测信息输出：



**图 7-3 测试验证**

经检测，信息输出正常。

经检测，只有输入正确密码后，才能进入系统，并且只有正确输入姓名后，才能显示对应信息。

3、可以通过增加冗余代码来反静态反汇编，通过计时、检测堆栈等来反跟踪。

7.2任务2

解密同组同学的加密程序，获取该同学的成绩。

7.2.1设计思想及存储单元分配

直接在TD进行静态反汇编，阅读反汇编得到的汇编代码。然后再TD单步调试，得到具体数据段的数据信息。

7.2.2流程图

7.2.3源程序

7.2.4实验步骤

1、反汇编exe程序，得到对应汇编代码。

2、静态阅读汇编代码，了解大致结构，找到反跟踪的部分。

3、在TD进行单步调试，进行跟踪破解。

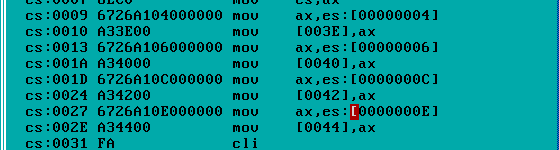
7.2.5 实验记录

1、实验环境条件：VM XP虚拟机，处理器: 2，内存：1GB；命令行窗口；Sublime Text 2；MASM.EXE 6.0； LINK.EXE 5.2; TD.EXE 5.0。

2、反汇编得到对应汇编程序：

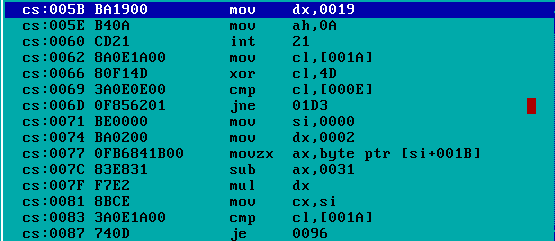
1. 查看代码，找到反跟踪的地方：

堆栈检查反跟踪



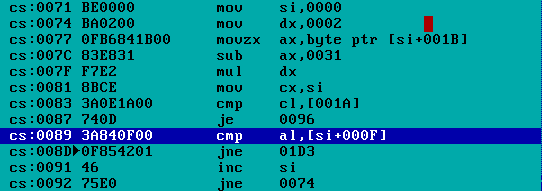
**图 7-4 堆栈检查反跟踪**

1. 单步调试：
2. 单步调试时要手动设置IP的值，从而跳过这些反跟踪的部分。
3. 单步调试时找到输入密码的地方，发现有对密码长度作比较的部分，然后直接根据反汇编代码，解密得到密码长度为7。



**图 7-5 获取密码长度**

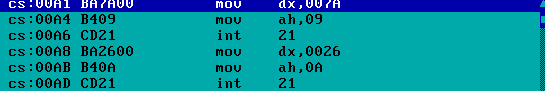
1. 找到对密码进行处理的地方，得到秘钥，对内存的密文解密，得到密码



**图7-6 解密密码**

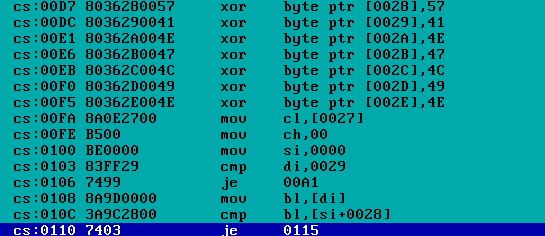
如图，可得秘钥为(x-0x31)\*2,则找到从000f开始的连续7个字节，解密的到密码为：WANGLIN。

4）单步调试，找到输入学生姓名的地方：



则缓冲区是从0026开始

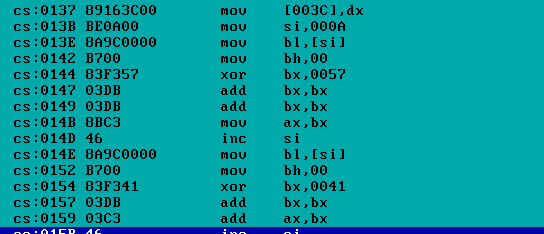
5）找到对输入进行处理的地方，获取秘钥，对内存中的密文解密：



**图 7-7 解密学生姓名**

如图，接的学生姓名为LENGWEI

6) 找到处理成绩的地方，直接观察寄存器bx的值，就可以得到对应成绩



**图 7-8 解密成绩**

如图，解密成绩得到对应成绩为：100，90，80，94

7）至此，解密完成。

3、如果密码明文存放，直接观察内存就能获得密码

4、通过静态反汇编代码了解代码结构，了解反跟踪的设置以及相应输入的缓冲区在内存的地址，然后动态单步调试跳过反跟踪，并观察秘钥，解密。

5、有反跟踪的地方一般都会有CLI和STI，直接跳过即可。

体会

通过本次上机，熟悉了对信息的加密、解密处理，了解怎样跟踪和反跟踪。

本次上机的具体收获有：（1）对内存中的信息进行加密处理。（2）设置反跟踪来防止别人窃取信息。(3) 通过反汇编来破解别人程序，从而获取想要的信息。

同时，对反汇编得到的代码分析和处理还不够熟悉，需要进一步学习。

# 参考文献

[序号] 作者.书名.版本(第×版).译者.出版地:出版者,出版年：起页-止页

[序号] URL: 网络地址，如URL:http://www.cbs.dtu.dk/services