

**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** WIN32程序设计

**实验时间： 2017-11-28，18：30-21：50 实验地点： 南一楼804室**

**指导教师： 朱虹**

**专业班级：信息安全 201602班**

**学 号： U201614836 姓 名： 林晓斌**

**同组学生： 无 报告日期： 2017年 11 月 28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：2017.10.17

|  |  |  |
| --- | --- | --- |
| 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

[实验六 win32编程 3](#_Toc500882314)

[6.1实验目的与要求 3](#_Toc500882315)

[6.2实验任务 3](#_Toc500882316)

[6.3实验内容 5](#_Toc500882317)

[6.3.1任务1 5](#_Toc500882318)

[6.3.2 任务2 7](#_Toc500882319)

[6.4实验小结 18](#_Toc500882320)

[参考文献 18](#_Toc500882321)

# 实验六 win32编程

# 6.1实验目的与要求

熟悉WIN32程序的设计和调试方法；

熟悉宏汇编语言中INVOKE、结构变量、简化段定义等功能；

进一步理解机器语言、汇编语言、高级语言之间以及实方式、保护方式之间的一些关系。

# 6.2实验任务

任务1 ：Win32程序开发奠基

熟悉Win32程序的汇编、链接和调试；熟悉Win32程序的结构和执行过程。

① 安装MASM32软件包，观察MASM32软件包目录结构和环境配置 （附件中有： INSTALL.exe）。

② 学习win32程序的编译和链接方法。(以 cbox 为例，附件中有： MASM32\_汇编链接的操作说明.doc)

③ 学习使用 VS2013 ( VS2010, VS2012都是类似的，附件中有： VS2013\_win7\_64 下汇编程序的编译链接.doc。 VS2013需要同学们自己安装)

（对于安装的 VC++6.0的同学，可以学习使用VC++ 6.0 来调试程序。附件中有：VC 6 下Win32汇编程序源码级调试.docx）

④ 阅读cbox的源程序，了解程序代码的特点和执行流程，体会基于窗口的应用程序所包含的四个部分之间的衔接关系。

⑤ 程序调试。观察程序模块间的调用关系；观察 Invoke等语句翻译成机器码后的结果、以及执行相应指令时堆栈的变化；观察执行一些指令前后，内存以及寄存器的变化；观察结构变量中数据的存放形式。

任务2 ：Win32编程

编写一个基于窗口的WIN32程序，实现学生成绩表信息的平均值计算及显示功能（借鉴前面实验中的一些做法），具体要求如下描述。

功能一：编写一个基于窗口的WIN32程序的菜单框架，具有以下的下拉菜单项：

File Action Help

Exit Average About

List

点菜单File下的Exit选项时结束程序；点菜单Help下的选项About，弹出一个消息框，显示本人信息，类似图1所示。点菜单Action下的选项Average、List将分别实现计算平均值或显示所有成绩的功能（详见功能二的描述）。



图1 菜单示例

功能二：每个学生的相关信息包括：姓名（结尾含1个以上的数值0，共占10个字节），语文成绩（1个字节），数学成绩（1个字节），英语成绩（一个字节），平均成绩（1个字节），等级（1个字节）。要求采用结构变量存放学生的相关信息。学生人数至少5人。姓名和各科成绩直接在数据段中给定，不必运行时输入。成绩表中最后一个学生必须使用自己的姓名。

点菜单项Average时，计算平均成绩并给出等级（等级的定义见实验一，但这里不用单独显示等级）。平均成绩的计算仍按照实验一的公式进行。平均成绩和等级保存到上述结构变量的相应字段中。

点菜单项List时，要求能在窗口中列出所有学生信息，包括姓名、各科成绩、平均成绩、等级等。如图2所示。平均成绩尚未计算时，平均成绩及等级显示为空白（调用API函数：TextOut）。

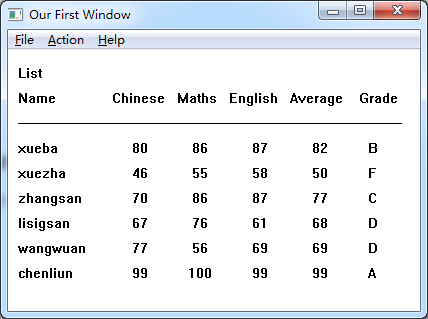


图2 成绩单显示示意图

任务3：选做题

1、增加程序的密码保护功能

在开始运行任务2的程序时，出现对话框，要求输入密码。若输入的密码错误，则给出提示信息，并等待再次输入密码。输入密码正确后，可正常执行原有功能。

2、密码破解

对上面的程序，在只有执行程序的情况下，找出密码。或者暴力修改执行程序，绕过对密码的检查。

# 6.3实验内容

6.3.1任务1

6.3.1.1程序运行截图

使用vs2013，按流程汇编如以下操作

(1)启动 VC++ 6.0

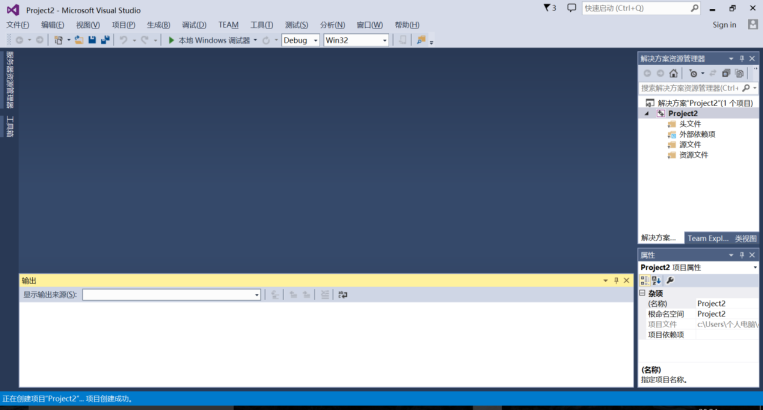
(2)打开待调试的执行文件

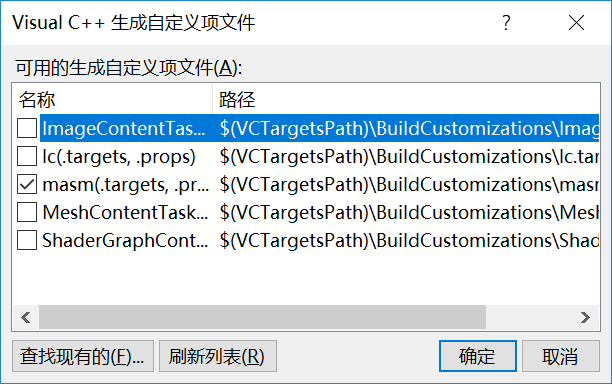
(3) 打开源程序

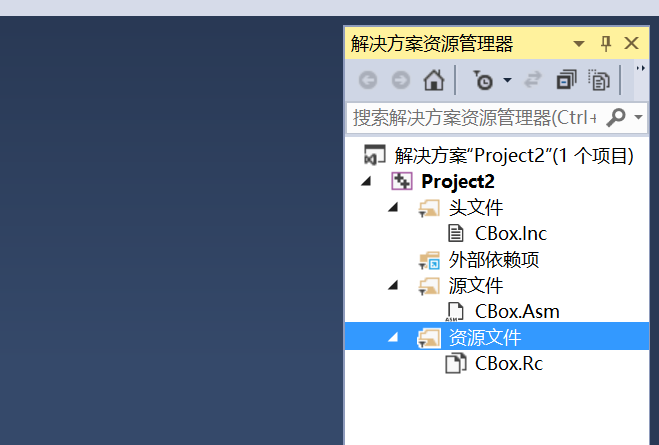
(4) 在源程序上设置断点

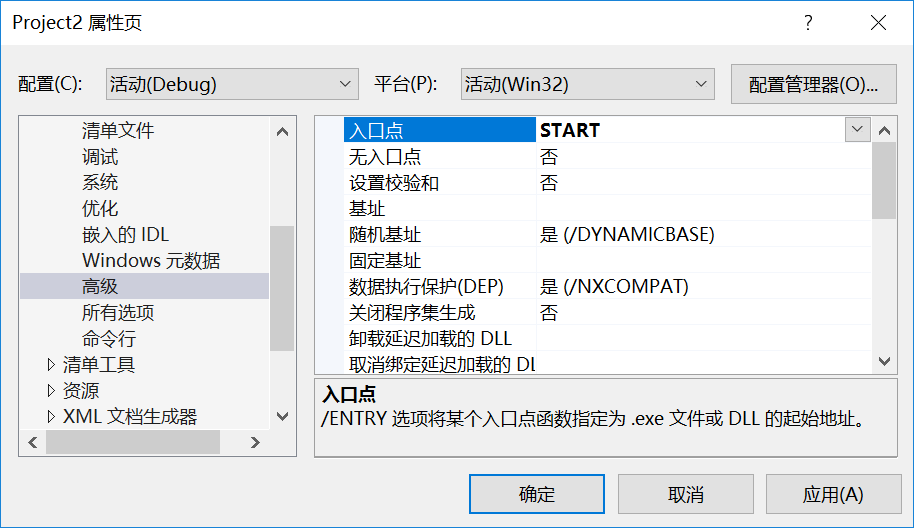
(5) 其他调试操作与 C语言程序的调试非常类似

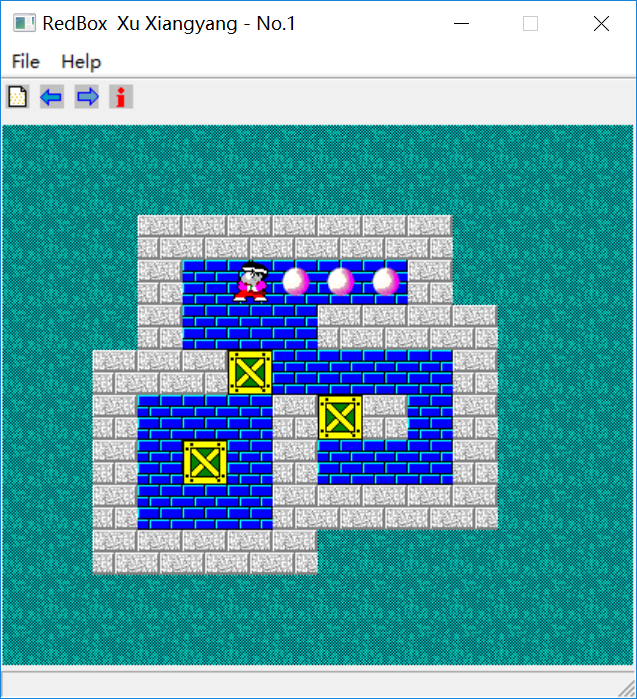
截图如下所示











6.3.2 任务2

6.3.2.1实验设计思路以及存储器的分配

编写一个基于windows窗口的程序，使其可以进行对学生信息的显示以及求学生平时成绩，最后对其成绩进行判定。其中，对于学生信息显示采用showlist子程序，对于求平时成绩以及成绩判定，合并在一起使用average子程序，窗口消息处理程序为winproc，窗口主程序为winmain，最后还附加上转化字符串的子程序。

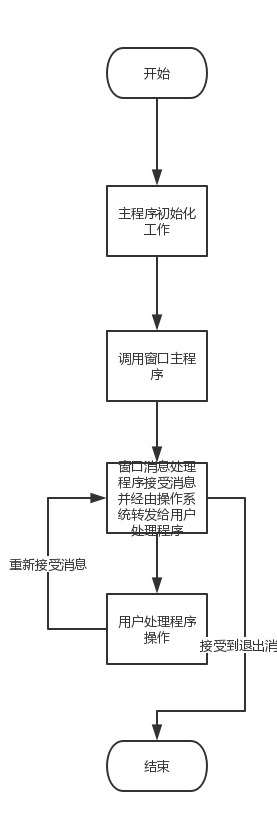
首先通过GetModuleHandle获得实例句柄，然后进入窗口主程序winmain。进入窗口主程序后，先填写RegisterClassex的参数WNDCLASSEX结构，然后通过RegisterClassex注册窗口类，再用CreateWindow建立窗口，保存窗口句柄后再通过ShowWindow显示窗口。之后通过LoadMenu根据资源标识符读入菜单的资源，再通过SetMenu装配到主窗口上，然后进入消息循环，通过Getmassage获取消息。

DispatchMessage回调窗口消息处理子函数WinProc，根据wParam的值来选择调用功能。标识符在rc文件中已经设定，然后通过massagebox在小窗口显示个人姓名，通过textout显示学生信息。

调用average可以对学生的成绩进行运算，并通过自己定义的string函数对其进行数字转字符串，然后将转化后的字符串（平均成绩等）放入显示的student结构体中。

最后通过exitprocess函数退出程序。

6.3.2.2 流程图



6.3.2.3 程序运行截图

先将主程序汇编之后再将rc资源程序汇编，然后将二者链接，形成可执行程序1.exe。程序运行效果如下图所示。

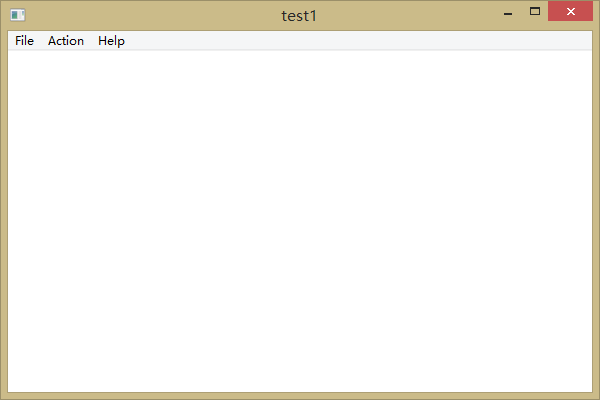


图3-1程序运行外观

点击file可弹出窗口exit，选择exit即退出程序，点开action选择list使学生信息打印在窗口上，平均成绩以及等级尚未显示，如图1-2所示。

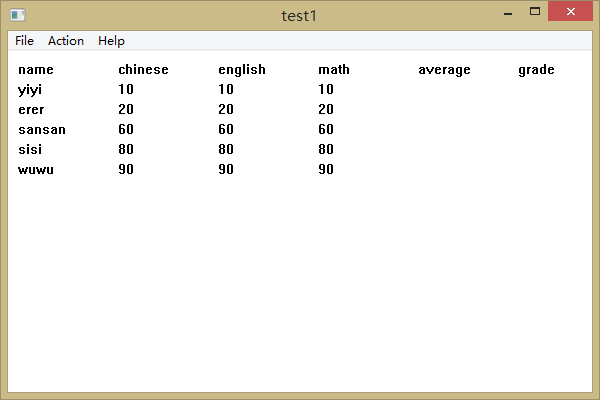


图3-2 打印学生信息

选择action中的average，计算学生的平均成绩，并为其判定程序，在选择action中的list，将其打印出来，如图3-3。

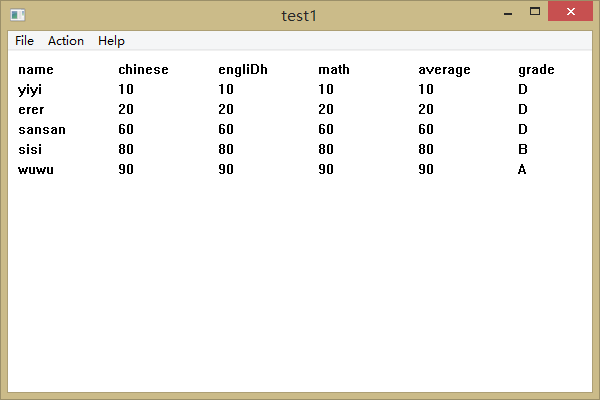


图3-3 显示平均成绩及等级

按下help中的about，可以显示个人信息，如图3-4.

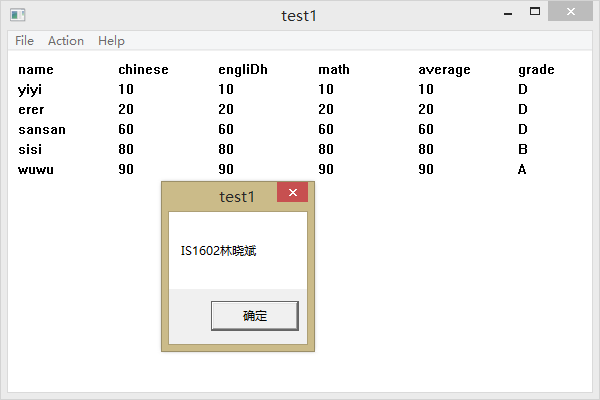


图3-4 个人信息显示

6.3.2.4 源程序

.386

.model flat, stdcall

option casemap :none

include windows.inc

include user32.inc

include kernel32.inc

include masm32.inc

include gdi32.inc

include comctl32.inc

includelib gdi32.lib

includelib user32.lib

includelib kernel32.lib

includelib masm32.lib

includelib comctl32.lib

WinMain proto:dword,:dword,:dword,:dword

WndProc proto:dword,:dword,:dword,:dword

TextOut proto:dword,:dword,:dword,:dword

string proto:dword,:dword

average proto

student struct

sname db 10 dup(' ')

chinese db 2 dup(' ')

english db 2 dup(' ')

math db 2 dup(' ')

average db 2 dup(' ')

grade db 1 dup(' ')

student ends

student1 struct

sname db 10 dup(0)

chinese db 0

english db 0

math db 0

average db 0

grade db 0

student1 ends

.data

szDisPlayName db "test1",0

szEditClass db "edit",0

MenuName db "test2",0

szClassName db "mainwndclass",0

aboutmsg db "IS1602林晓斌",0

hInstance dd 0

CommandLine dd 0

hWnd dd 0

buf1 student1<'yiyi',10,10,10,90,>

buf2 student1<'erer',20,20,20,90,>

buf3 student1<'sansan',60,60,60,90,>

buf4 student1<'sisi',80,80,80,90,>

buf5 student1<'wuwu',90,90,90,90,>

buf6 student<'yiyi','10','10','10',' ',' '>

buf7 student<'erer','20','20','20',' ',' '>

buf8 student<'sansan','60','60','60',' ',' '>

buf9 student<'sisi','80','80','80',' ',' '>

buf0 student<'wuwu','90','90','90',' ',' '>

msg\_name db 'name',0

msg\_chinese db 'chinese',0

msg\_english db 'english',0

msg\_math db 'math',0

msg\_average db 'average',0

msg\_grade db 'grade',0

.CODE

START:

invoke GetModuleHandle,NULL

mov hInstance,eax

INVOKE GetCommandLine

mov CommandLine,eax

invoke WinMain,hInstance,NULL,CommandLine,SW\_SHOWDEFAULT

invoke ExitProcess,eax

;-----------------------main progress-----------

WinMain proc hInst :dword,

hPrevInst :dword,

Cmdline :dword,

Cmdshow :dword

local wc:WNDCLASSEX

local msg1:MSG

local wwd:DWORD

local wht:DWORD

local wtx:DWORD

local wty:DWORD

local rectclient:RECT

MOV wc.cbSize,sizeof WNDCLASSEX

mov wc.style,CS\_HREDRAW + CS\_VREDRAW

mov wc.lpfnWndProc,offset WndProc

mov wc.cbClsExtra,NULL

mov wc.cbWndExtra,NULL

push hInst

pop wc.hInstance ;实例句柄赋值

mov wc.hbrBackground,COLOR\_WINDOW+1 ;color

mov wc.lpszMenuName,offset MenuName

mov wc.lpszClassName,offset szClassName

mov wc.hIcon,0

invoke LoadCursor,NULL,IDC\_ARROW

mov wc.hCursor,eax

mov wc.hIconSm,0

invoke RegisterClassEx,addr wc

mov wwd,600

mov wht,400

mov wtx,10

mov wty,10

invoke CreateWindowEx,WS\_EX\_ACCEPTFILES+WS\_EX\_APPWINDOW,addr szClassName,addr szDisPlayName,WS\_OVERLAPPEDWINDOW+WS\_VISIBLE,wtx,wty,wwd,wht,NULL,NULL,hInst,NULL ;创建可视窗口（hInst好像没定义）

mov hWnd,eax

invoke LoadMenu,hInst,600

invoke SetMenu,hWnd,eax

;消息循环

StartLoop: invoke GetMessage,addr msg1,NULL,0,0

cmp eax,0

je ExitLoop

invoke TranslateMessage,addr msg1

invoke DispatchMessage,addr msg1

jmp StartLoop

ExitLoop:

mov eax,1000

ret

WinMain endp

;用户处理程序

;按行打印msg\_，buf数据段表示方式有待考量

showlist proc

LOCAL hdc: HDC

push eax

invoke GetDC,hWnd

mov hdc,eax

invoke TextOut,hdc,10,10,addr msg\_name,4

invoke TextOut,hdc,110,10,addr msg\_chinese,7

invoke TextOut,hdc,210,10,addr msg\_english,7

invoke TextOut,hdc,310,10,addr msg\_math,4

invoke TextOut,hdc,410,10,addr msg\_average,7

invoke TextOut,hdc,510,10,addr msg\_grade,5

invoke TextOut,hdc,10,30,ADDR buf6.sname,10

invoke TextOut,hdc,110,30,ADDR buf6.chinese,2

invoke TextOut,hdc,210,30,ADDR buf6.english,2

invoke TextOut,hdc,310,30,ADDR buf6.math,2

invoke TextOut,hdc,410,30,ADDR buf6.average,2

invoke TextOut,hdc,510,30,ADDR buf6.grade,1

invoke TextOut,hdc,10,50,ADDR buf7.sname,10

invoke TextOut,hdc,110,50,ADDR buf7.chinese,2

invoke TextOut,hdc,210,50,ADDR buf7.english,2

invoke TextOut,hdc,310,50,ADDR buf7.math,2

invoke TextOut,hdc,410,50,ADDR buf7.average,2

invoke TextOut,hdc,510,50,ADDR buf7.grade,1

invoke TextOut,hdc,10,70,ADDR buf8.sname,10

invoke TextOut,hdc,110,70,ADDR buf8.chinese,2

invoke TextOut,hdc,210,70,ADDR buf8.english,2

invoke TextOut,hdc,310,70,ADDR buf8.math,2

invoke TextOut,hdc,410,70,ADDR buf8.average,2

invoke TextOut,hdc,510,70,ADDR buf8.grade,1

invoke TextOut,hdc,10,90,ADDR buf9.sname,10

invoke TextOut,hdc,110,90,ADDR buf9.chinese,2

invoke TextOut,hdc,210,90,ADDR buf9.english,2

invoke TextOut,hdc,310,90,ADDR buf9.math,2

invoke TextOut,hdc,410,90,ADDR buf9.average,2

invoke TextOut,hdc,510,90,ADDR buf9.grade,1

invoke TextOut,hdc,10,110,ADDR buf0.sname,10

invoke TextOut,hdc,110,110,ADDR buf0.chinese,2

invoke TextOut,hdc,210,110,ADDR buf0.english,2

invoke TextOut,hdc,310,110,ADDR buf0.math,2

invoke TextOut,hdc,410,110,ADDR buf0.average,2

invoke TextOut,hdc,510,110,ADDR buf0.grade,1

pop eax

ret

showlist endp

string proc bufa:DWORD, bufb:DWORD

push eax

push dx

push di

PUSH EBX

MOV EAX,bufb

mov al,BYTE ptr[EAX]

mov ah,0

mov dx,0

MOV DI,10

DIV DI

ADD AL,30H

MOV EBX,bufa

MOV BYTE ptr [EBX],AL

ADD DL,30H

MOV BYTE ptr [EBX+1],DL

POP EBX

pop di

pop dx

pop eax

ret

string endp

average proc

push ax

push bx

push ecx

push esi

push edi

mov bl,3

xor ax,ax

mov al,buf1.chinese

add al,buf1.english

add al,buf1.math

div bl

mov buf1.average,al

invoke string,addr buf6.average,addr buf1.average

mov al,buf2.chinese

add al,buf2.english

add al,buf2.math

div bl

mov buf2.average,al

invoke string,addr buf7.average,addr buf2.average

mov al,buf3.chinese

add al,buf3.english

add al,buf3.math

div bl

mov buf3.average,al

invoke string,addr buf8.average,addr buf3.average

mov al,buf4.chinese

add al,buf4.english

add al,buf4.math

div bl

mov buf4.average,al

invoke string,addr buf9.average,addr buf4.average

mov al,buf5.chinese

add al,buf5.english

mov ah,0

mov cl,buf5.math

mov ch,0

add ax,cx

div bl

mov buf5.average,al

invoke string,addr buf0.average,addr buf5.average

mov ecx,0

mov esi,0

mov edi,0

lp: cmp buf1[edi].average,90

jge level1

cmp buf1[edi].average,80

jge level2

cmp buf1[edi].average,70

jge level3

jmp level4

level1: mov al,'A'

mov BYTE ptr buf6[esi].grade,al

jmp quit

level2: mov al,'B'

mov BYTE ptr buf6[esi].grade,al

jmp quit

level3: mov al,'C'

mov BYTE ptr buf6[esi].grade,al

jmp quit

level4: mov al,'D'

mov BYTE ptr buf6[esi].grade,al

jmp quit

quit: inc ecx

add esi,19

add edi,15

cmp ecx,5

jle lp

pop edi

pop esi

pop ecx

pop bx

pop ax

ret

average endp

;窗口消息处理程序

WndProc proc hWin :dword,

uMsg :dword,

wParam :dword,

lParam :dword

.if uMsg == WM\_COMMAND

.if wParam ==3000

invoke showlist

.elseif wParam ==2000

invoke average

.elseif wParam == 4000

invoke MessageBox,hWin,addr aboutmsg,addr szDisPlayName,MB\_OK

.elseif wParam ==1000

invoke ExitProcess,0

.endif

.elseif uMsg ==WM\_DESTROY

invoke PostQuitMessage,NULL

.else

invoke DefWindowProc,hWin,uMsg,wParam,lParam

ret

.endif

mov eax,0

ret

WndProc endp

;暂停显示,回车键关闭

invoke ExitProcess,0

end START

# 6.4实验小结

本次实验难度感觉比较大，因为新接触了一种比较陌生的编程，对于为win32编程一开始并没有什么概念，编程的时候也是无从下手，特别是许许多多没见过的函数和他们的参量，都需要对照书一步一步做，还有一些全新的概念待了解（句柄等）。但是当自己真正写完运行程序时，即使是一个小小的窗口也让人感到兴奋，终于摆脱了黑乎乎的运行环境。对于win32编程，它功能复杂，一方面可以丰富他可实现的功能，另一方面其实也加大了编程者的掌握难度。但是win32编程的实用程度高，以后还得多啃啃。

# 参考文献

[1] 王元珍、曹忠升、韩宗芬.80X86汇编语言程序设计(第一版).华中科技大学出版社，2005年:1-348

[2] 许向阳.80X86汇编语言程序设计上机指南(第一版).华中科技大学出版社,2007年:1-224