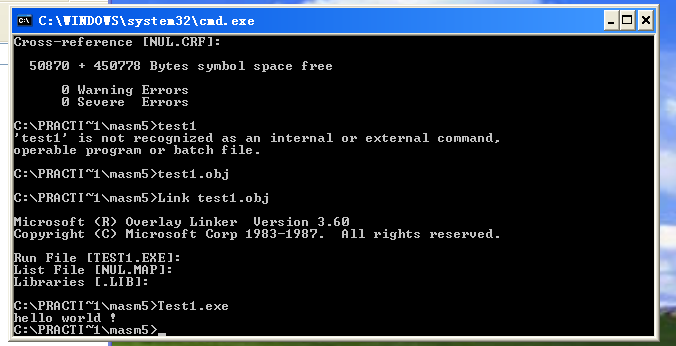
# 系统级编程Lab03

姓名：袁豪

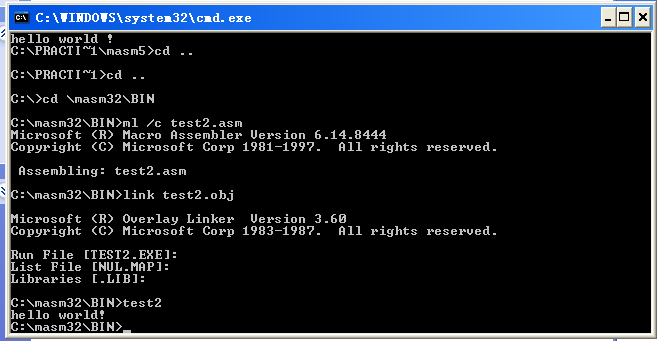
学号：2014141463269

备注：主要实验在VM上的32位XP系统上完成的

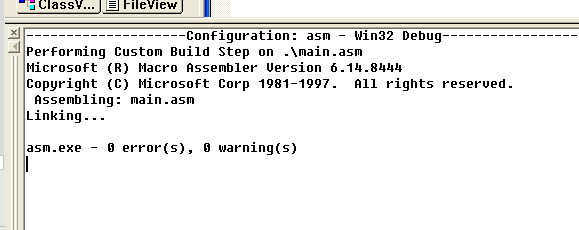
# Practice1. masm5.X 开发一个完整段汇编程序/16位

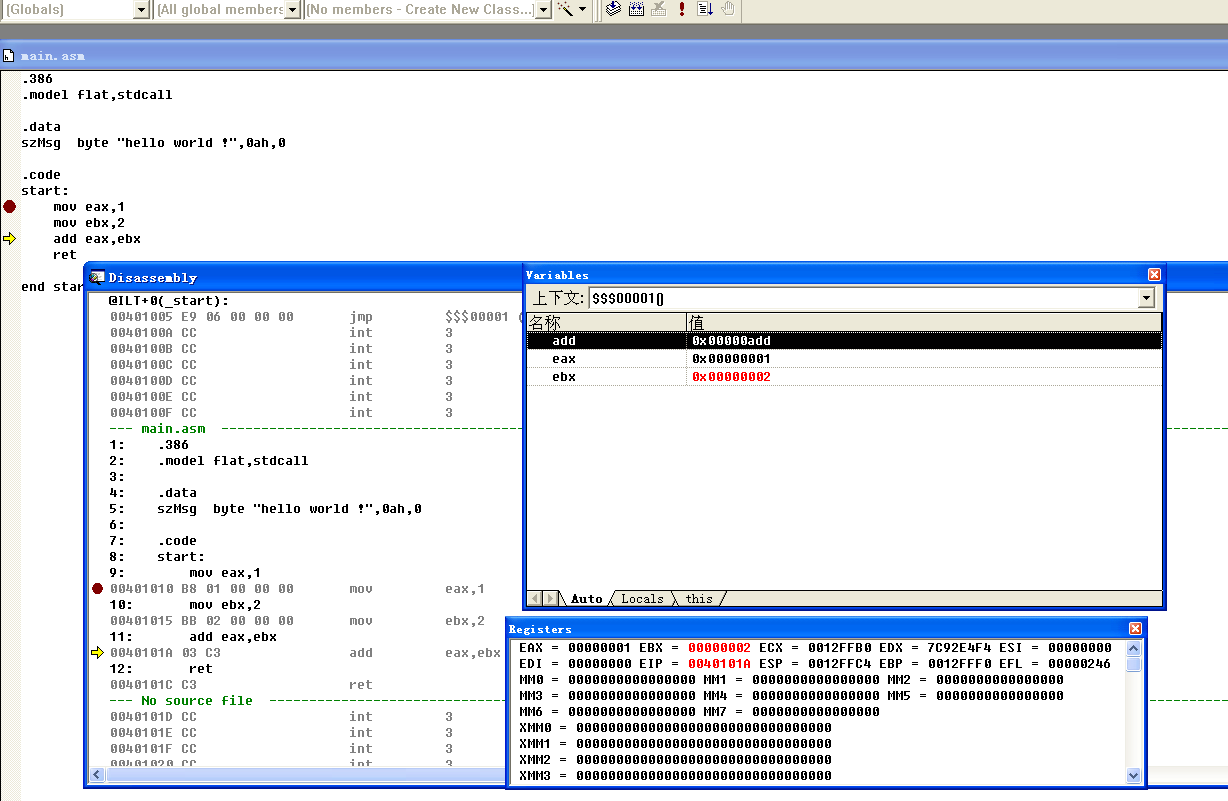


# Practice2. masm6.X 开发一个简化段汇编程序/16位

、

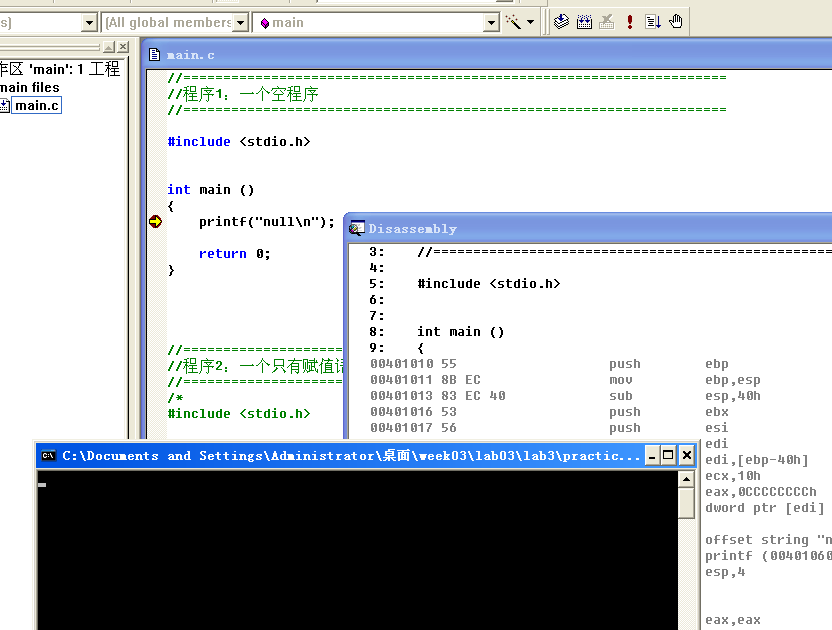
# Practice3. VC++ 6.0 IDE 开发一个汇编程序/32位



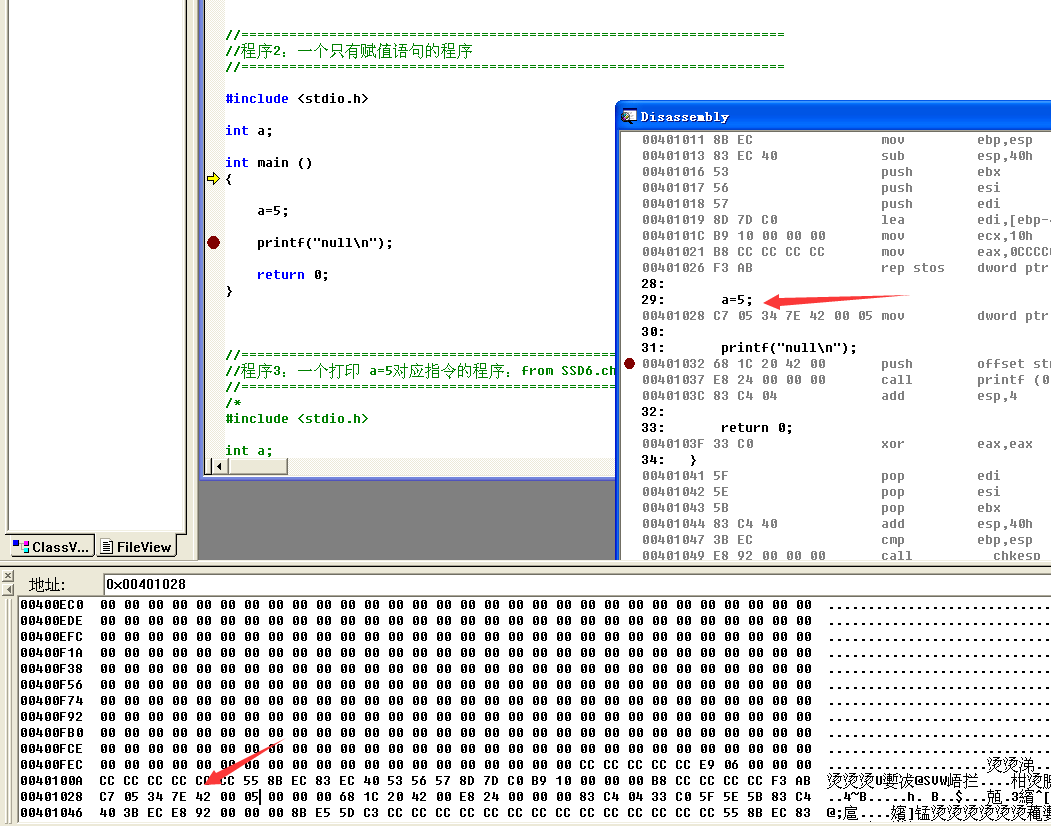


# Practice4. VC++ 6.0 IDE 反汇编C程序

## 1. 一个空程序

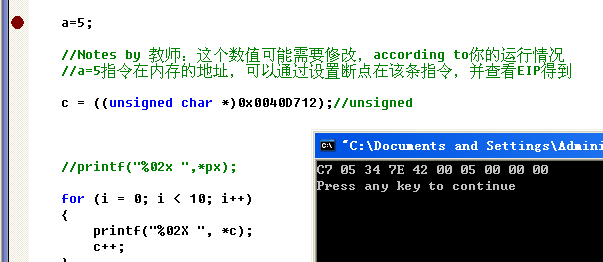


## 2. 一个只有赋值语句的程序



## 3．一个只有赋值语句的程序

经调试，发现a=5所在内存地址为0x0040D712，故修改代码并运行如下



.(1)输出：C7 05 34 7E 42 00 05 00 00 00

（2）含义：前面的C7 05是操作码，也就是执行的指令；34 7E 42 00表示操作数，即a,最后四位表示操作数5。

（3）如何做到的：程序定义了指向a=5指令所在的内存地址的指针\*C，通过指针的移动来打印数据十位操作数和操作码。

## 4. 一个带有算术运算的程序

汇编代码:

0040D708 A1 6C 51 42 00 mov eax,[\_c (0042516c)]

0040D70D 6B C0 03 imul eax,eax,3

0040D710 8B 0D 68 51 42 00 mov ecx,dword ptr [\_b (00425168)]

0040D716 03 C8 add ecx,eax

0040D718 89 0D 34 7E 42 00 mov dword ptr [\_a (00427e34)],ecx

指令：

Mov, imul, add

寄存器：

Eax, acx

## 5. 一个带有位运算的程序

00401028 A1 30 4A 42 00 mov eax,[\_a (00424a30)]

0040102D 23 05 34 4A 42 00 and eax,dword ptr [\_b (00424a34)]

00401033 A3 58 7C 42 00 mov [\_c (00427c58)],eax

## 6. 一个带有逻辑运算的程序

0040D718 83 3D 30 4A 42 00 00 cmp dword ptr [\_a (00424a30)],0

0040D71F 74 12 je main+33h (0040d733)

0040D721 83 3D 34 4A 42 00 00 cmp dword ptr [\_b (00424a34)],0

0040D728 74 09 je main+33h (0040d733)

0040D72A C7 45 FC 01 00 00 00 mov dword ptr [ebp-4],1

0040D731 EB 07 jmp main+3Ah (0040d73a)

0040D733 C7 45 FC 00 00 00 00 mov dword ptr [ebp-4],0

0040D73A 8B 45 FC mov eax,dword ptr [ebp-4]

* 0040D73D A3 58 7C 42 00 mov [\_c (00427c58)],eax

## 7. 一个if分支的程序，回答问题：

00401032 83 3D 30 7E 42 00 00 cmp dword ptr [\_a (00427e30)],0

00401039 7E 0D jle main+38h (00401048)

总结：

先执行if要括号里要判断的条件，再执行一个jump(JNE, JLE等),jump语句后面带有偏移地址。

## 8. 一个if分支的程序，回答问题：

00401032 cmp dword ptr [\_a (00427e30)],3

00401039 jle main+38h (00401048)

先执行if要括号里要判断的条件，再执行一个jump(JNE, JLE等),jump语句后面带有偏移地址。

## 9. 一个if-else分支的程序

If:

00401032 83 3D 30 7E 42 00 00 cmp dword ptr [\_a (00427e30)],0

00401039 7E 0F jle main+3Ah (0040104a)

Else:

00401048 EB 0D jmp main+47h (00401057)

总结：If结构同7，else执行指令jmp，跳转到相应语句

## 10. 一个switch分支的程序

0040D722 mov eax,[\_a (00427e30)]

0040D727 mov dword ptr [ebp-4],eax

0040D72A cmp dword ptr [ebp-4],1

0040D72E je main+3Eh (0040d73e)

0040D730 cmp dword ptr [ebp-4],2

0040D734 je main+4Ah (0040d74a)

0040D736 cmp dword ptr [ebp-4],3

0040D73A je main+54h (0040d754)

0040D73C jmp main+60h (0040d760)

总结：先将用于判断的变量放入累加器，再将eax放入一个指针，对于变量可能的情况分别用cmp语句判断，每种相应的情况就jump到对应的语句。

## 11. 一个while循环的程序

0040D722 cmp dword ptr [\_a (00427e30)],0

0040D729 jle main+45h (0040d745)

0040D743 jmp main+22h (0040d722)

框架：先执行cmp, 如果满足条件则执行括号里的部分。括号里代码执行完后再jump回while所在语句再判断，如果不满足条件则执行下一句jle，跳出循环。

## 12. 一个do-while循环的程序

0040D73A cmp dword ptr [\_a (00427e30)],0

0040D741 jg main+22h (0040d722)

框架：先执行循环部分，当一次循环结束后进行判断，如果满足条件则jg回循环开始的地方，不满足则跳出循环。

## 13. 一个for循环的程序

0040D722 mov dword ptr [\_a (00427e30)],5

0040D72C jmp main+3Bh (0040d73b)

0040D72E mov eax,[\_a (00427e30)]

0040D733 sub eax,1

0040D736 mov [\_a (00427e30)],eax

0040D73B cmp dword ptr [\_a (00427e30)],0

0040D742 jle main+55h (0040d755)

319: {

320: b++;

0040D744 mov ecx,dword ptr [\_b (00427e34)]

0040D74A add ecx,1

0040D74D mov dword ptr [\_b (00427e34)],ecx

321: }

0040D753 jmp main+2Eh (0040d72e)

框架：先对变量进行赋值，赋值结束后跳转到判断语句，满足条件则开始执行循环，每次循环结束后又jump回对eax的操作语句（a--），又重复进行判断和循环。

# Practice5. 开发一个简单的x86 BootLoader汇编程序

代码：

org 07c00h ; 告诉编译器程序加载到7c00处

mov ax, cs

mov ds, ax

mov es, ax

call DispStr ; 调用显示字符串例程

jmp $ ; 无限循环

DispStr:

mov ax, BootMessage ;将BootMessage的值传递给ax

mov bp, ax ；将ax的值传给bp

mov cx, 16 ;cx值即显示的字符串长度，此处设置为16

mov ax, 01301h ；旋转AH = 13的显示方式

mov bx, 000ah

;第二步设置dos中断调用，以显示字符串

int 10h ; 10h 号中断

ret

;下面代码无需改动

BootMessage: db "Hello, OS world!"

times 510-($-$$) db 0 ; 填充剩下的空间，使生成的二进制代码恰好为512字节

dw 0xaa55 ; 结束标志

运行效果如图：

