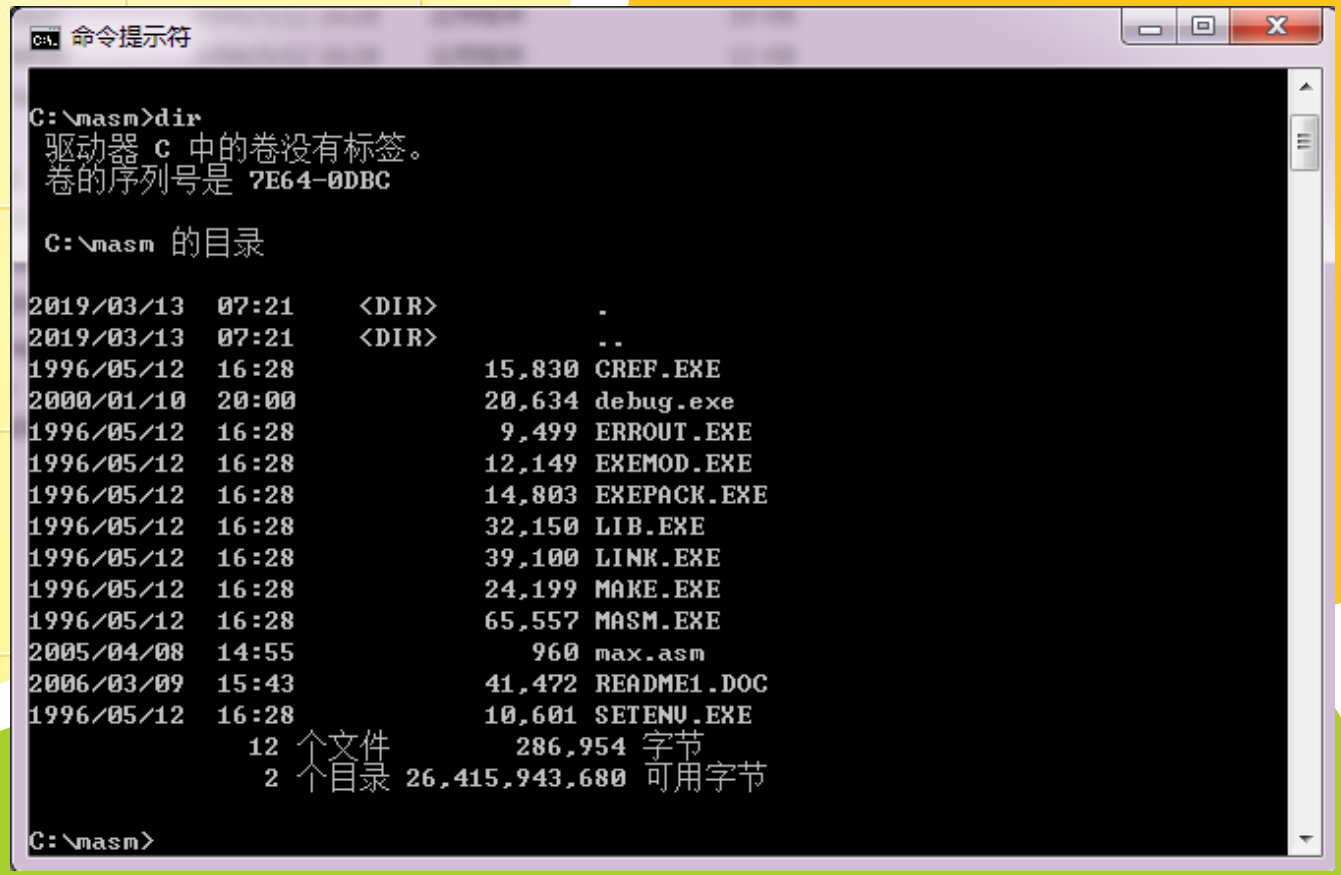
1、X86汇编语言程序调试环境
2、DEBUG调试工具

masm	2017/3/1 10:22	压缩(zipped)文件...	193 KB
DOSBox0.74-win32-installer	2019/3/13 7:24	应用程序	1,415 KB

- 汇编开发调试软件——DOS命令行方式 masm
- DOS模拟器DOSBox的安装和使用
- 调试环境DEBUG



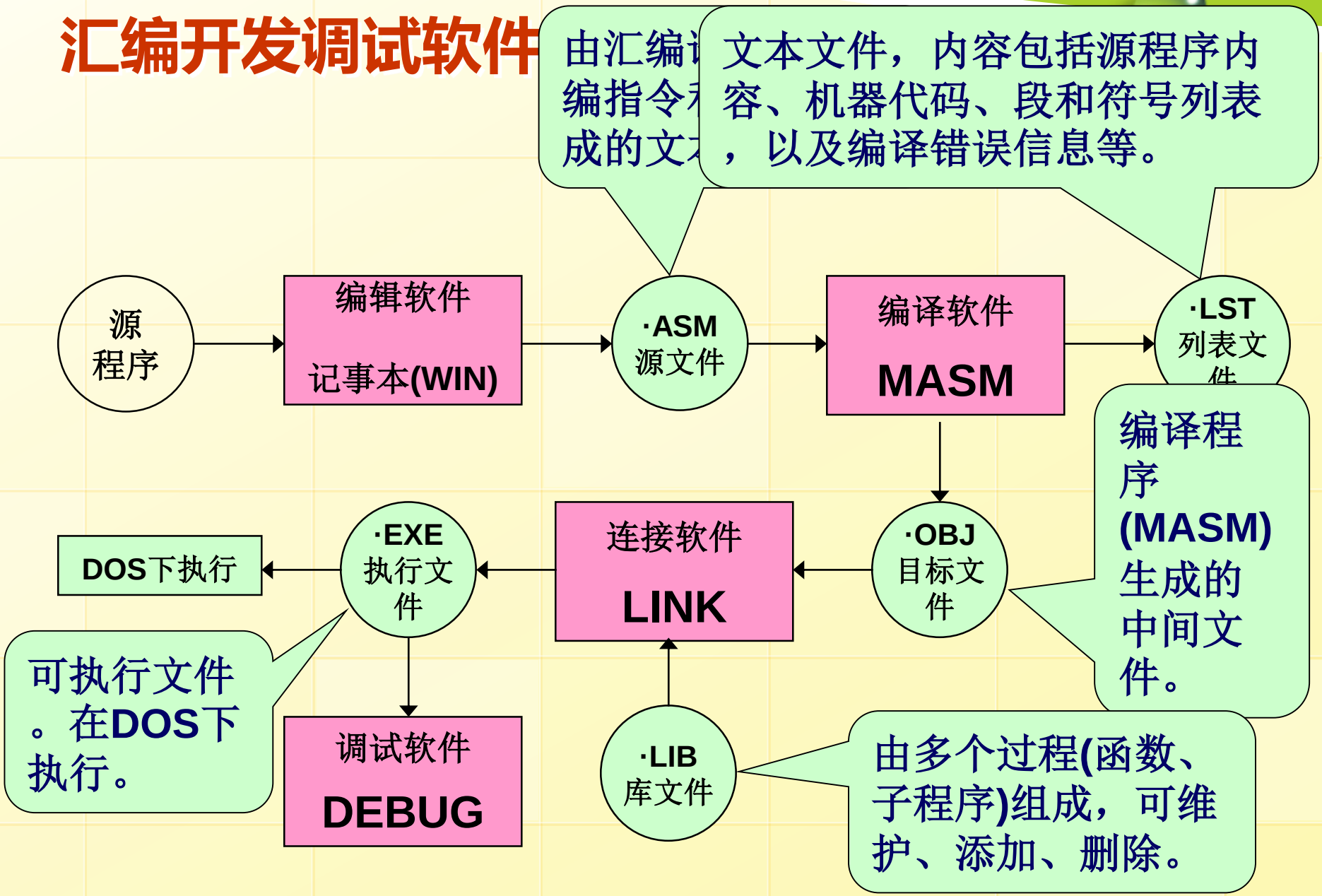
汇编开发调试软件——DOS命令行方式

X86汇编语言使用的开发调试工具包括汇编源程序的编译、连接，以及动态调试。

1.用x86 汇编语言的开发环境MASM,进行源代码的编译和连接，形成执行文件.

2.使用DEBUG进行执行代码的动态调试和简单汇编指令的执行。

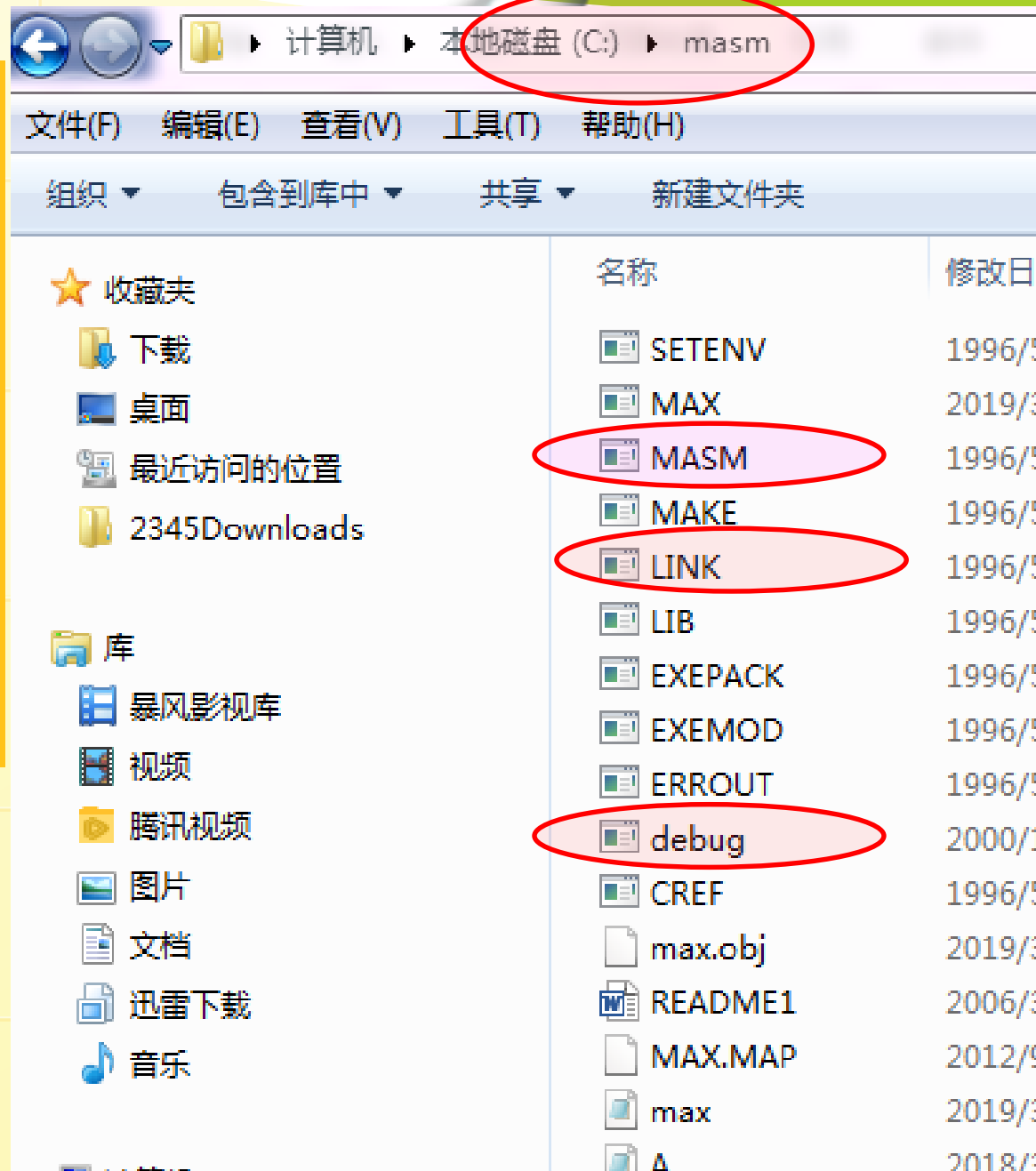
汇编开发调试软件


由汇编
编指令
成的文
文本文件，内容包括源程序内容、机器代码、段和符号列表，以及编译错误信息等。
源程序
编辑软件
记事本(WIN)
·ASM
源文件
编译软件
MASM
·LST
列表文件
编译程序(MASM)生成的中间文件。
·OBJ
目标文件
连接软件
LINK
·EXE
执行文件
DOS下执行
可执行文件。在DOS下执行。
调试软件
DEBUG
·LIB
库文件
由多个过程(函数、子程序)组成，可维护、添加、删除。

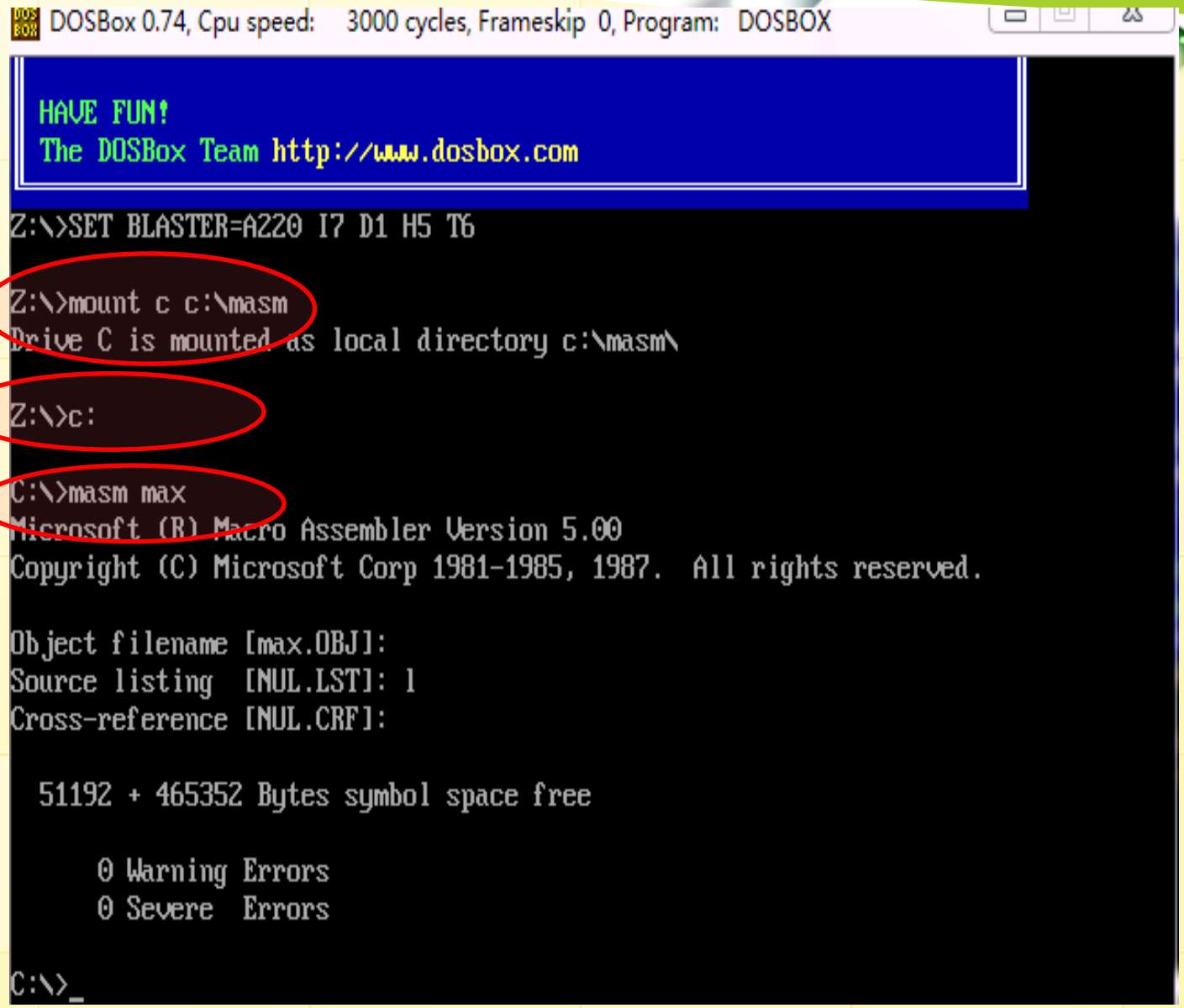
DOS模拟器DOSBox的安装和使用

在Windows里直接运行DOS程序经常会产生问题，特别是装有64位操作系统的计算机，不能运行汇编程序调试环境MASM下的执行文件，需要使用免费的DOS模拟器DOSBox。

1. 在下面的官网下载dosBOX：
http://www.dosbox.com/download.php?main=1
2.把masm文件夹放到一个盘的根目录下，以便将MASM挂载到DOSBox，这里MASM 在c:\目录下
3、执行DOSBox文件



4. 在DOSBox控制台界面下输入：
mount c c:\masm
其中:mount 是挂载命令。此命令的作用是将本地目录 c:\masm 挂载在DOSBox控制台的C盘下，这样可以在DOSBox控制台下运行DOS程序，也就是可以运行MASM下的编译命令masm ，链接命令link 和调试命令DEBUG。
右图表示把c:\masm虚拟挂载在一个C:\目录下，然后输入：c:(表示打开c:\，进入DOSBOX控制的环境下)
5. 之后就可以使用MASM下的MASM，LINK和DEBUG命令进行汇编程序的调试了。



汇编程序的开发MASM的使用

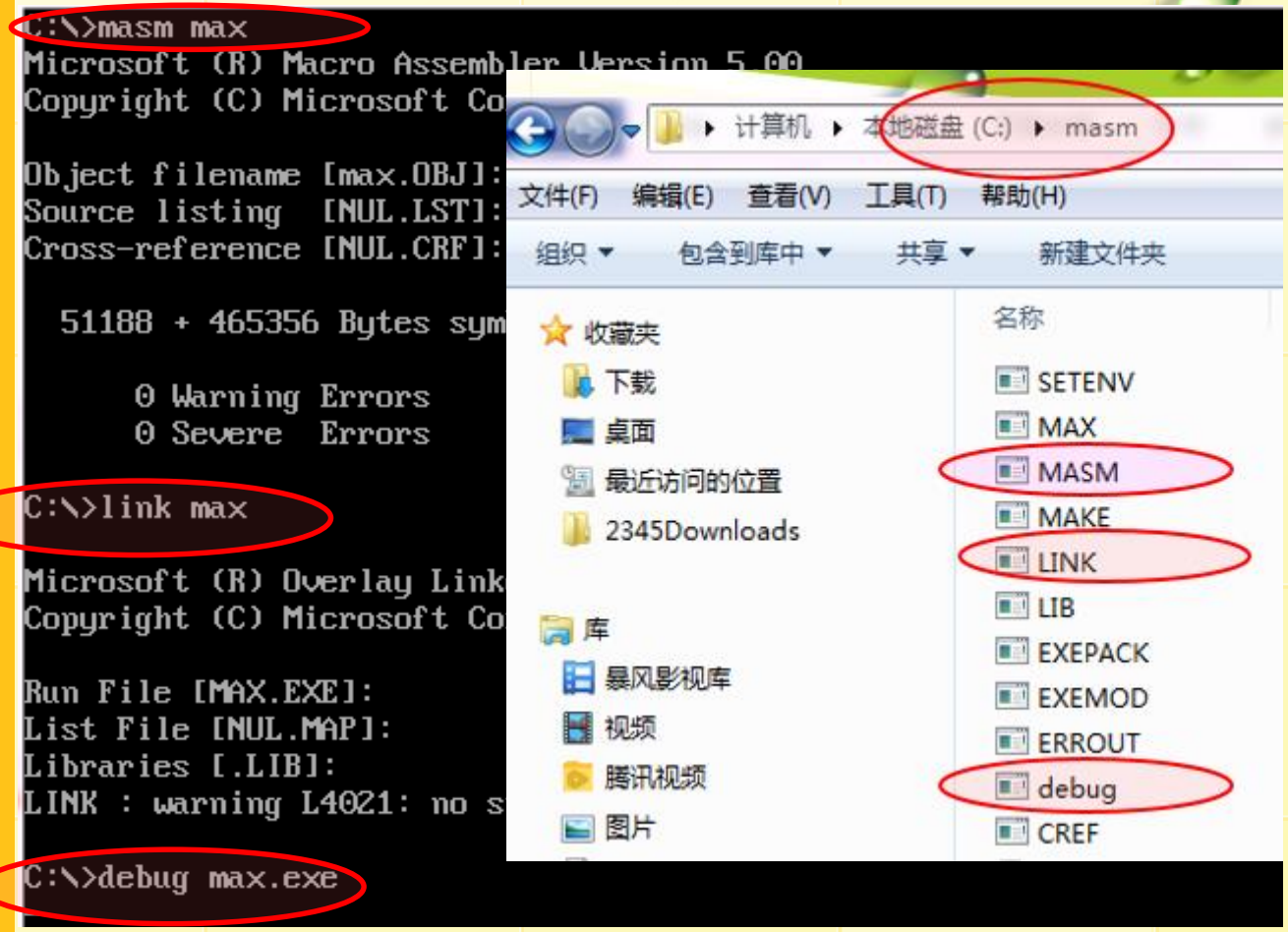
需要编辑、编译和连接-MASM完成和调试（DEBUG 完成），步骤：

1、利用写字板等编辑软件，编辑一个X86汇编程序，比如求三个数最大值的源程序MAX.ASM

2、利用MASM进行编译，MASM MAX.ASM(文件后缀可有可无)，查看编译后的列表文件MAX.LST 文件，可以发现源代码的错误所在，及其错误的原因等，直到编译通过后，才能进行下一步的目标文件的链接LINK；

3、利用LINK进行链接，，即LINK MAX.OBJ(文件后缀可有可无)，生成执行文件MAX.EXE。

4、调入执行程序到内存，准备调试：即键入 c:\masm>debug max.exe（文件后缀必须有)



DEBUG调试工具介绍

DEBUG是DOS操作系统提供的程序之一，在DOS提示符下键入程序名“DEBUG”，屏幕上出现的短划线是“DEBUG”的提示符，这符号通知用户机器已作好准备来接受下一个命令。

```
C:\>debug max.exe
-_
```

DEBUG命令

1.“ D ”命令

功能:显示内存单元的内容.

注意:十六进制是DEBUG唯一认识的数字系统,键入和输出时十六进制数不用后加“H”,十进制数后要加“D”

格式:

-D [存储单元地址]

例如:

-D 7A0:100

显示的信息是段地址是7A0H和偏移地址是100H开始的内存单元的内容，表示方式是一行16个数据，然后是对于的ASCII字符。

DEBUG命令

2.“ E ”命令

输入数据后按空格键
可连续修改数据

功能:修改存储单元的内容

格式:

-E 存储单元地址 数据 数据 数据 ...

此格式使键入的数据替代了指定范围的存储单元内容

-E 存储单元地址

此格式采用逐个存储单元相继修改的方法

DEBUG命令

3.“ F ”命令

功能:用一个指定的十六进制数填入一部分存储单元

格式:

-F 开始地址 终止地址 常量

“F”命令与“E”命令都可修改存储单元的内容,但“F”命令只能键入一串相同的数据。

例如:

-F 100 120 4F

DEBUG命令

4.“ R ”命令

功能:检查和修改寄存器的内容

格式:

(1) -R

执行:显示CPU内所有寄存器内容和标志位状态

(2) -R 寄存器名字

执行:显示和修改某个寄存器的内容

(3) -RF

执行:显示和修改标志位状态

DEBUG命令

5.“ A ”命令

功能:汇编命令

格式:

-A [开始存放指令的地址]

可以键入汇编语言语句,并能把它汇编成机器代码,相继放入从指定地址开始的存储器中，回车键退出，回到DEBUG提示符下.

DEBUG命令

6.“T”命令

功能:跟踪命令

格式:

(1) -T [=指令地址]

功能:从指定的地址起执行一条指令后停下来,显示所有寄存器内容及标志位的值;如未指定地址则从当前的CS:IP开始执行

(2) -T [=指令地址][要执行的指令条数N]

功能:从指定地址开始执行N条指令后停下来,显示

DEBUG命令

7.“ G ”命令

功能：运行命令

格式：

-G=<地址>，<断点>

比如：

-G=0，1C

执行:从起始地址0开始运行,在断点地址1C处停止,并显示指令从0开始，到断点1C执行后，所有寄存器及标志位内容,以及下一条要执行的指令.

Program Status Word

	A	B	C	D
1		Psw中的标志位的符号表示		
2	分类	标志名	标志为1	标志为0
3	状态标志：用来反映EU执行算术和逻辑运算以后的结果特征。	OF 溢出标志	OV	NV
4		ZF零标志	ZR	NZ
5		AF辅助进位标志	AC	NA
6		PF奇偶标志	PE偶数	PO奇数
7		CF进位标志	CY	NC
8		SF 符号标志	为正数	为负数
9	控制标志：用来控制CPU的工作方式或工作状态。	TF：陷阱标志或单步操作标志SP	NG	PL
10		IF中断标志	EI允许中断	DI
11		DF方向标志	DN	UP
12				

```
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1397  ES=1397  SS=1397  CS=1397  IP=0100   NV UP EI PL NZ NA PO NC
1397:0100 3132          XOR     [BP+SI],SI                    SS:0000=20CD
-R AX
AX 0000
:1050
-R
AX=1050  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=1397  ES=1397  SS=1397  CS=1397  IP=0100   NV UP EI PL NZ NA PO NC
1397:0100 3132          XOR     [BP+SI],SI                    SS:0000=20CD
```