**一. DOSBox简介**

DOSBox 是一个 DOS 模拟程序，由于它采用的是 SDL 库，所以可以很方便的移植到其他的平台。目前，DOSBox 已经支持在 Windows、Linux、Mac OS X、BeOS、palmOS、Android、webOS、os/2等系统中运行。

官方网址：<http://www.dosbox.com/>

DOSBox官网下载地址：https://www.dosbox.com/download.php?main=1

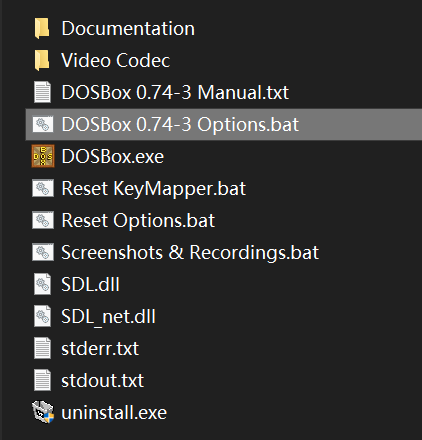
使用教程：https://www.dosbox.com/wiki/Basic\_Setup\_and\_Installation\_of\_DosBox

**二. 安装DOSBox**

安装：双击安装文件即可完成，并将安装目录中MASM5的相关编辑、汇编和连接文件拷贝到指定文件夹，如c:\mams5等。

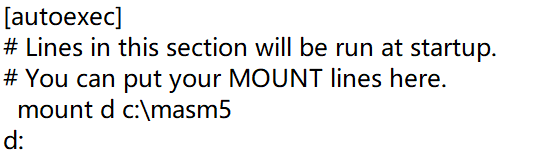
卸载：双击安装路径D:\Program Files (x86)\DOSBox-0.74中的uninstall.exe即可

**三. 配置DOSBox**

进入**DOSBox**的安装后文件夹并运行参数配置批处理文件。

将屏幕显示方式改为全屏模式，否则屏幕太小不利于调试。

每次调试程序时，都要先mount一下才能加载MASM5文件目录，比较麻烦，因此建议将参数改为如下配置，这样每次进入后会自动加载指定的文件目录。

其中c:\mams5为宏汇编MASM5的文件目录，d为映射盘符。

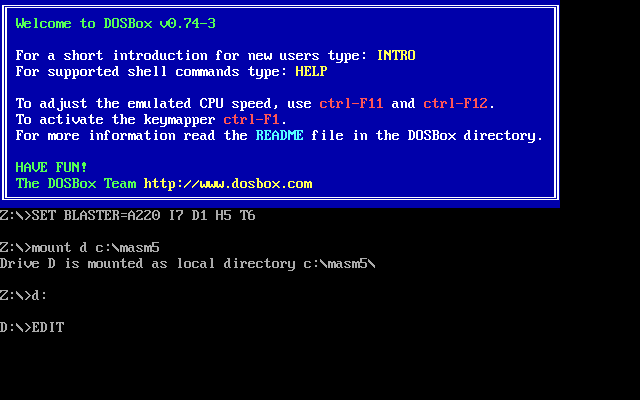
**四. 编辑edit.com**

可以用任意的文本编辑器来编辑源程序，只要最终将其存储为.asm纯文本文件即可。

当然也可以采用自带的edit.com程序进行汇编语言源代码程序的编辑。

运行DOSBOX，在“D:\>”提示符后输入edit，回车

注意：edit.com程序必需位于加载目录之中，如"C:\MASM5\"目录;



输入汇编代码：

assume cs:codesg

codesg segment

        mov ax,0123h

        mov bx,0456h

        add ax,bx

        add ax,ax

        mov ax,4c00h

        int 21h

codesg ends

end

File -> Save As…  弹出另存为对话框,在 File Name: 后面输入\*.asm，点击OK按钮，程序将被保存在缺省文件中，实际也就是C:\MASM5文件夹中。

**五. 编译MASM.exe**

对 \*.asm文件进行编译，生成包含机器代码的目标文件(.obj文件)。

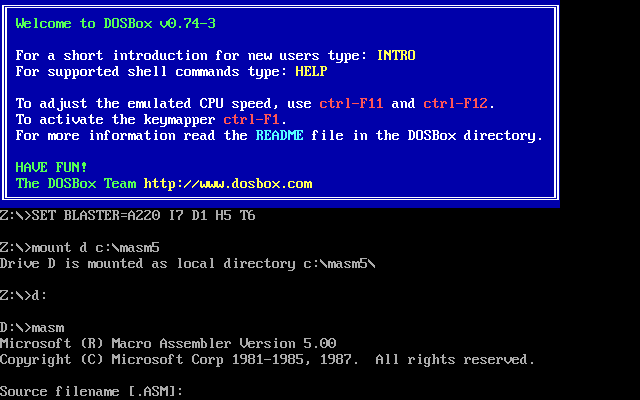
采用微软的masm5.0汇编编译器，文件名为masm.exe

在“D:\>”提示符后输入masm，回车

提示输入将要被编译的源文件的名称。

注意：

“[.ASM]”提示我们，默认的文件扩展名是asm，比如，要编译的源程序文件名是“p1.asm”，只要在这里输入“p1”即可。可如果源程序文件不是以asm为扩展名的话，就要输入它的全名。比如源程序文件名为“p1.txt”，就要输入全名。

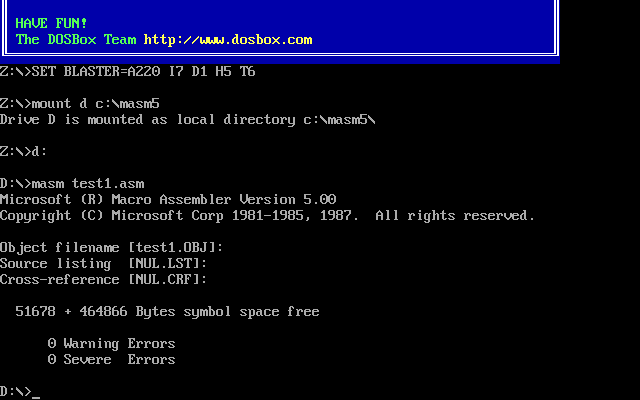
在输入源程序文件名的时候一定要指明它所在的路径。如果文件就在当前路径下，只输入文件名就可以，可如果文件在其他的目录中，则要输入路径，比如，要编译的文件p1.txt在“c:\windows\desktop”下，则要输入“c:\windows\desktop\p1.txt”。

  输入要编译的源程序文件名后，按Enter键。

在输入源程序文件名后，程序继续提示我们输入要编译出的目标文件的名称，目标文件是我们对一个源程序进行编译要得到的最终结果。注意下图屏幕上的显示：“[test1.obj]”我们已经输入了源程序文件名为test1.asm，则编译程序默认要输出的目标文件名为test1.obj，所以可以不必再另行指定文件名。直接按Enter键，编译程序将在当前的目录下，生成test1.obj文件。

  这里，也可以指定生成的目标文件所在的目录，比如，想让编译程序在“c:\windows\desktop”下生成目标文件1.obj，则可输入“c:\windows\desktop\1”。

确定了目标文件的名称后，屏幕显示如下图所示。



编译程序提示输入列表文件的名称，这个文件是编译器将源程序编译为目标文件的过程中产生的中间结果。可以让编译器不生成这个文件，直接按Enter键即可。

编译程序提示输入交叉引用文件的名称，这个文件同列表文件一样，是编译器将源程序编译为目标文件过程中产生的中间结果。可以让编译器不生成这个文件，直接按Enter键即可。

对源程序的编译结束，编译器输出的最后两行告诉我们这个源程序没有警告错误和必须要改正的错误。

按照上面的过程进行了编译之后，在编泽器masm.exe运行的目录C:\masm5下(即当前路径下)，将出现一个新的文件：test1.obj目标文件。

**六. 连接LINK.EXE**

对 \*.obj目标文件进行连接，得到可执行文件(.exe文件)。

采用微软的LINK.EXE

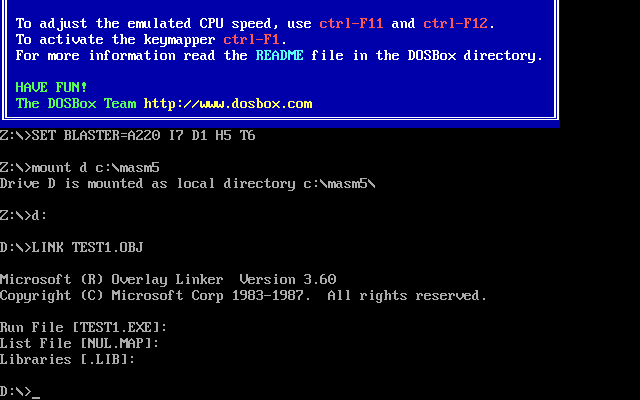
在“D:\>”提示符后输入link，回车

  运行link后，首先显示一些版本信息，然后提示输入将要被连接的目标文件的名称。

注意，“[.OBJ]”提示我们，默认的文件扩展名是obj。

在输入目标文件名的时候，要注意指明它所在的路径。这里，要连接的文件是当前目录下的test1.obj。

在输入目标文件名后，程序继续提示我们输入要生成的可执行文件的名称，可执行文件是我们对一个程序进行连接要得到的最终结果。注意屏幕上的显示：“[test1.EXE]”，因为已经确定了源程序文件名为test1.obj，则程序默认要输出的可执行文件名为test1.EXE，所以可以不必再另行指定文件名。直接按Enter键，编译程序将在当前的目录下，生成test1.EXE文件。

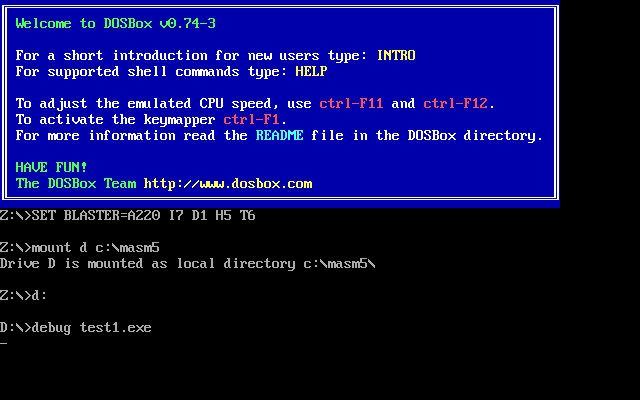
也可以指定生成的可执行文件所在的目录，比如，想让连接程序在“c:\windows\desktop”下生成可执行文件 1.EXE，则可输入“c:\windows\desktop\1”。我们直接按Enter键，使用连接程序设定的可执行文件名。

连接程序提示输入映像文件的名称，这个文件是连接程序将目标文件连接为可执行文件过程中产生的中间结果，可以让连接程序不生成这个文件，直接按Enter键即可。

连接程序提示输入库文件的名称。库文件里而包含了一些可以调用的子程序，如果程序中调用了某一个库文件中的子程序，就需要在连接的时候，将这个库文件和目标文件连接到一起，生成可执行文件。但是，这个程序中没有调用任何子程序，所以，这里忽略库文件名的输入，直接按Enter键即可。

按照上面的过程进行了连接之后，在连接器link.exe运行的目录c:\masm5下(即当前路径下)，将出现一个新的文件：test1.exe，这是对目标文件test1.obj进行连接所得到的结果。当然，如果连接过程中出现错误，那么将得不到可执行文件。

**七. 程序执行过程的跟踪**

在“D:\>”提示符后输入DEBUG \*.exe，回车

 DEBUG的主要命令

**(1)、检查和修改寄存器内容的命令R，它有三种方式：**

1）显示CPU内部所有寄存器内容和标志位状态，格式为：-R

例如：输入-R↙，得到结果如下：

 2）显示和修改某个指定寄存器内容，格式为：-R 寄存器名

例如：输入-R AX，得到结果如下：

 若不对其作修改，可按ENTER键，否则，输入修改内容后，如：

则AX内容由0000改为0A0A

3）R命令显示标志寄存器标志位状态的含义如下表所示：

|  |  |  |
| --- | --- | --- |
| 标 志 名 | 置  位（值为1） | 复  位（值为0） |
| 溢出Overflow（是/否） | OV | NV |
| 方向Direction（减量/增量） | DN | UP |
| 中断Interrupt（允许/屏蔽） | EI | DI |
| 符号Sign（负/正） | NG | PL |
| 零Zero（是/否） | ZR | NZ |
| 辅助进位Auxiliary Carry（是/否） | AC | NA |
| 奇偶Parity（偶/奇） | PE | PO |
| 进位Carry（是/否） | CY | NC |

修改标志位状态，命令格式为：-RF

例如：输入-RF，输出结果如下：

 这时若不作修改可按ENTER键，否则在“-”号之后键入修改值，键入顺序任意。

**(2)、汇编命令A，格式为：-A[地址]**

该命令从指定地址开始允许输入汇编语句，把它们汇编成机器代码相继存放在从指定地址开始的存储器中。A命令中如果没有指定地址，则接着上一个A命令的最后一个单元开始；若还没有使用过A命令，则从当前CS : IP开始。

例如：输入代码，则代码存储在OAFO:0100至OAFO:010D单元中。

**(3)、反汇编命令U，有两种格式：**

1）-U[地址]

该命令从指定地址开始，反汇编32个字节，若地址省略，则从上一个U命令的最后一条指令的下一个单元开始显示32个字节。

 2）-U范围

该命令对指定范围的内存单元进行反汇编，例如：

-U 0100 011E或-U 100 11E 或-U 0AF0：0100 011E或–U 0AF0：0100 L1F（L用来引导指令的条数，1F为指令的条数），这些命令是等效的。

**(4)、运行命令G，格式为：**

-G [=地址1][地址2]

其中地址1规定了运行起始地址，后面的地址为断点地址。若省略地址，则运行从IP所指的地址开始。

**(5)、单步调试命令T，有两种格式：**

1）逐条指令追踪：

-T

该命令从指定地址起执行一条指令后停下来，显示寄存器内容和状态值。

 2）多条指令追踪：

-T[=地址][值]

该命令从指定地址起执行n条命令后停下来，n由[值]确定。

**(6)、显示内存单元内容的命令D，格式为：**

-D[地址]（从给出的地址开始连续显示128个字节的单元内容）或-D[范围]。

**(7)、修改内存单元内容的命令E，它有两种格式：**

1）用给定的内容代替指定范围的单元内容：

-E地址  内容表

 例如：-E 2000：0100  F3 “XYZ” 8D

其中F3，“X”“Y”“Z”和8D各占一个字节，用这五个字节代替原内存单元2000：0100到0104的内容，“X”“Y”“Z”将分别按它们的ASCII码值代入。

 2）逐个单元相继地修改：

-E地址

例如：-E 100：

0AF0：0100 F3.E7

此命令是将原100号单元的内容F3改为E7。E7是键入值。

**(8)、退出DEBUG命令Q，该命令格式为：**

-Q

退出DEBUG程序，返回DOS。