

**汇编语言与逆向技术课程实验报告**

**实验一：HelloWorld**

****

学 院 网络空间安全学院

专 业 信息安全

学 号 2211044

姓 名 陆皓喆

班 级 信息安全

**一、实验目的**

1、熟悉Win32汇编MASM32的编译环境；

2、命令行输出“Hello world！”；

3、在窗口输出“Hello world！”。

**二、实验原理**

MASM32是国外的MASM爱好者自行整理和编写的一个软件包，可以用于编译。MASM32汇编编译器是MASM6.0以上版本中的ml.exe，资源编译器是Microsoft Visual Studio中的rc.exe，32位链接器是Microsoft Visual Studio中的Link.exe，同时包含有其他的一些如lib.exe和DumpPe.exe等工具。

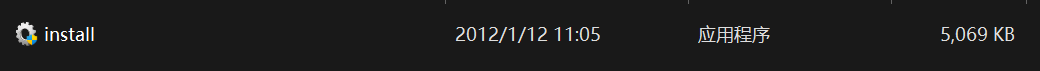
本次实验的主要原理是运用电脑中的cmd命令提示符来进行文本的输出，其中命令提示符需要用到MASM32中的一些文件。

通过命令行的处理，我们可以在命令行和窗口输出“Hello world！”。

**三、实验过程**

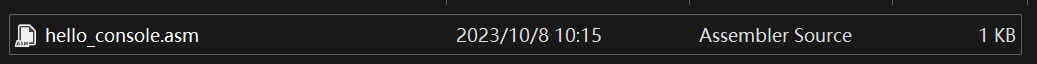
1.安装

根据老师下发的文件夹，安装MASM32。



2.导入

将提供的代码导入到文本框中，并且将文件后缀改为“.asm”。

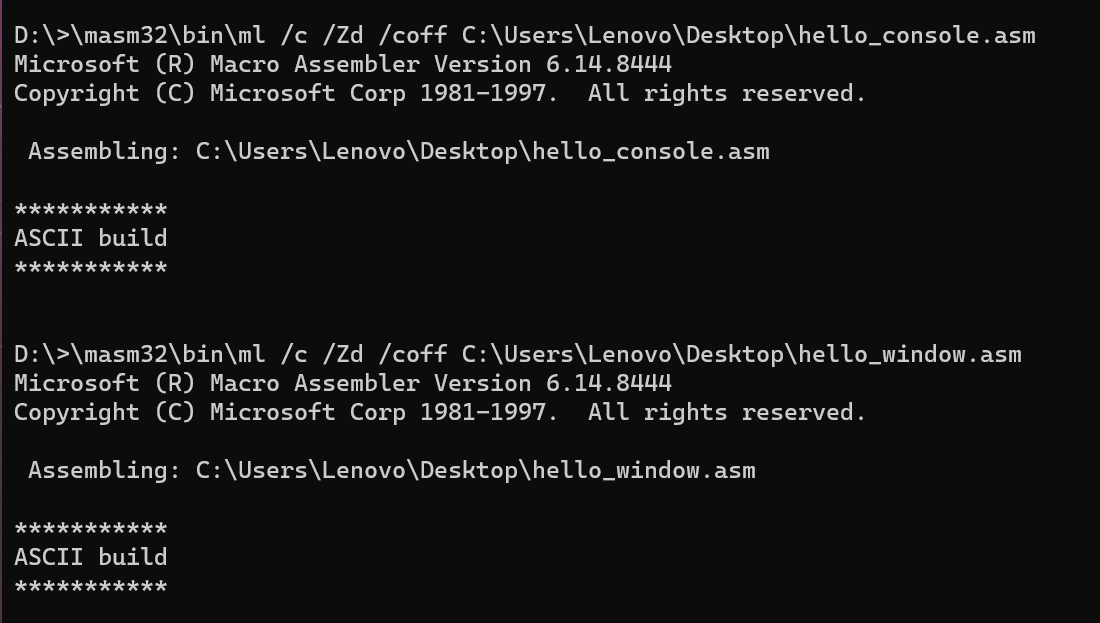




3.编译

对文件进行编译，在cmd命令提示符中进行操作。分别输入“\masm32\bin\ml /c /Zd /coff C:\Users\Lenovo\Desktop\hello\_console.asm”和“\masm32\bin\ml /c /Zd /coff C:\Users\Lenovo\Desktop\hello\_window.asm”即可得到文件。注意，在前面需要添加文件的地址，不然就会报错。

下面是在cmd命令行得到的结果：

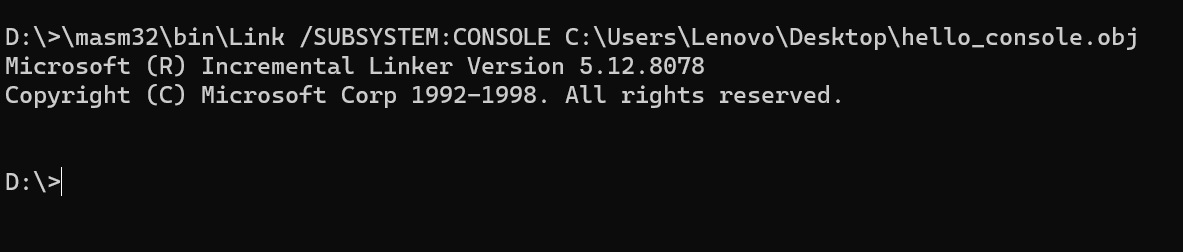


以下是经过命令行操作得到的两个obj文件：

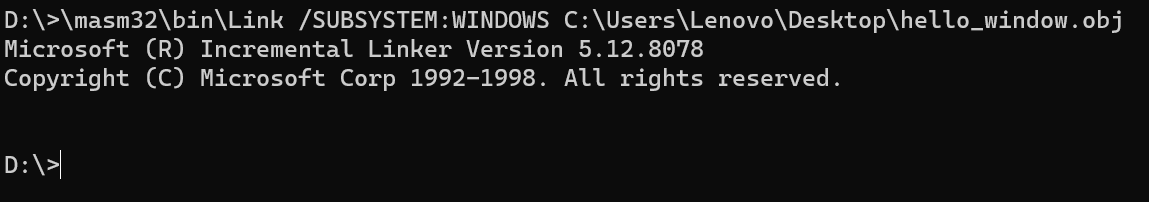




4. 连接

用连接程序（\masm32\bin\link.exe）对目标程序进行连接，形成可执行文件（.exe）。使用cmd命令行操作，分别输入：“\masm32\bin\Link/SUBSYSTEM:CONSOLE hello\_console.obj”和“\masm32\bin\Link /SUBSYSTEM:WINDOWS hello\_window.obj”即可得到最后的结果：两个exe文件。

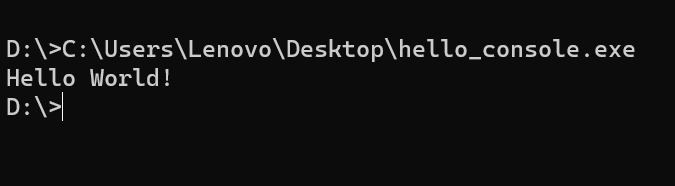






5.执行exe程序

第一个程序需要拖入cmd命令行，然后回车即可得出结果。附上结果图：



可以发现成功的输出了“Hello world！”。

第二个程序可以双击打开，双击后得到以下结果，附上结果图：



我们发现在windows窗体也成功的输出了“Hello world！”。

**四、实验代码解释**

**1．汇编命令和参数的解析**

**（1）\masm32\bin\ml /c /Zd /coff hello\_console.asm**

\masm32\bin\：**代表了masm32文件夹中的bin文件夹。**

\ml：**程序可以用来汇编并链接一个或多个汇编语言源文件，ml的命令行选项是大小写敏感的。**

/c：**Assemble without linking，只编译、不链接。**

/Zd： **Add line number debug info，在目标文件中生成行号信息。**

/coff： **generate COFF format object file，生成Microsoft公共目标文件格式（common object file format）的文件。**

hello\_console.asm：**是文件的名字。**

**（2）\masm32\bin\link /SUBSYSTEM:CONSOLE hello\_console.obj**

\masm32\bin\：**代表了masm32文件夹中的bin文件夹。**

\link：**链接器，将obj文件合并，生成可执行文件。**

/SUBSYSTEM:CONSOLE：**生成命令行程序。**

hello\_console.obj：**文件名。**

**2.汇编程序解析**

**（1）程序1：**

.386**（允许汇编80386处理器的非特权指令，禁用其后处理器引入的汇编指令）**

.model flat, stdcall**（初始化程序的内存模式，选择平坦模式，stdcall是Win 32 API 函数的调用约定）**

option casemap :none**（不区分大小写）**

include \masm32\include\windows.inc**（函数的常量和声明）**

include \masm32\include\kernel32.inc**（函数的常量和声明）**

include \masm32\include\masm32.inc**（函数的常量和声明）**

includelib \masm32\lib\kernel32.lib**（链接\masm32\lib\kernel32库）**

includelib \masm32\lib\masm32.lib**（链接\masm32\lib\masm32 库）**

.data**（定义已初始化数据段的开始）**

str\_hello BYTE "Hello World!", 0**（打印出字符串“Hello world！”）**

.code**（定义代码段的开始）**

start:**（指令标号，标记指令地址）**

invoke StdOut, addr str\_hello**（StdOut是masm32.inc中定义的函数，将内存数据输出到命令行窗口上）**

invoke ExitProcess, 0**（ExitProcess是Kernel32.inc中定义的函数，退出程序执行）**

END start**（标记模块的结束，指定程序的入口）**

**（2）程序2：**

.386**（允许汇编80386处理器的非特权指令，禁用其后处理器引入的汇编指令）**

.model flat, stdcall **（初始化程序的内存模式，选择平坦模式，stdcall是Win 32 API 函数的调用约定）**

option casemap :none **（不区分大小写）**

include \masm32\include\windows.inc **（函数的常量和声明）**

include \masm32\include\kernel32.inc **（函数的常量和声明）**

include \masm32\include\user32.inc **（函数的常量和声明）**

includelib \masm32\lib\kernel32.lib **（链接\masm32\lib\kernel32库）**

includelib \masm32\lib\user32.lib **（链接\masm32\lib\user32 库）**

.data**（定义已初始化数据段的开始）**

str\_hello BYTE "Hello World!", 0 **（打印出字符串“Hello world！”）**

.code **（定义代码段的开始）**

start: **（指令标号，标记指令地址）**

invoke MessageBox, NULL, addr str\_hello, addr str\_hello, MB\_OK **（MessageBox是将该字符串打印到控制台窗口上）**

invoke ExitProcess, 0**（ExitProcess是Kernel32.inc中定义的函数，退出程序执行）**

END start**（标记模块的结束，指定程序的入口）**

五、实验结论以及心得体会

通过这次实验，我初步了解了汇编MASM32的编译环境与环境的搭建，对汇编基础代码“hello word！”的文本框输出与命令行输出有了基本的了解。

但是初学对于各个代码还是有些一头雾水，希望以后能够有更加深刻的学习，能够更加深入地了解代码的含义。