**汇编语言与逆向技术实验报告**

**Lab1-HelloWorld**

**学号：2112060 姓名：孙蕗 专业：信息安全**

1. **实验内容**

本实验提供一个在命令行输出“HelloWorld”字符串的汇编程序，和一个在Windows MessageBox中输出“HelloWorld”的汇编程序。

1. **实验目的**
2. 熟悉Win32汇编MASM32的编译环境；
3. 命令行输出“HelloWorld”
4. 窗口输出“HelloWorld”
5. **实验原理**
   1. **MASM32**

MASM32是国外的MASM爱好者自行整理和编写的一个软件包，最高版本为11.0版，MASM32并不是[微软](https://baike.baidu.com/item/%E5%BE%AE%E8%BD%AF" \t "_blank)官方发布的软件，微软官方发布的软件MASM最新版本也只到6.15版，微软发布的MASM系列版本从6.11版才开始支持windows[编程](https://baike.baidu.com/item/%E7%BC%96%E7%A8%8B" \t "_blank)，6.11版以前的版本都不支持windows编程，只能用来写DOS程序。

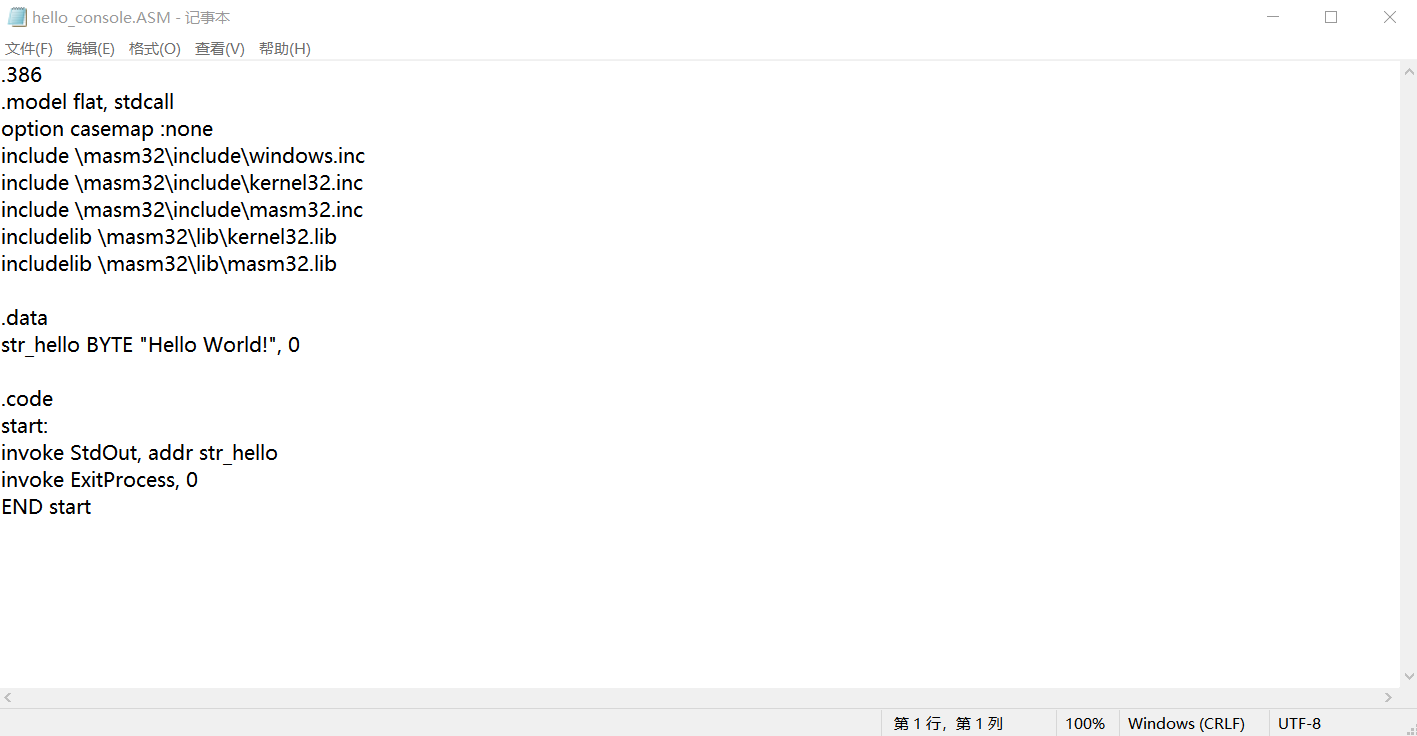
MASM32[汇编](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96" \t "_blank)[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8" \t "_blank)是MASM6.0以上版本中的ml.exe，资源编译器是[Microsoft Visual Studio](https://baike.baidu.com/item/Microsoft%20Visual%20Studio" \t "_blank)中的rc.exe，32位[链接器](https://baike.baidu.com/item/%E9%93%BE%E6%8E%A5%E5%99%A8" \t "_blank)是[Microsoft Visual Studio](https://baike.baidu.com/item/Microsoft%20Visual%20Studio" \t "_blank)中的Link.exe，同时包含有其他的一些如lib.exe和DumpPe.exe等工具。

MASM的windows编程的教学书籍有《windows环境下32位[汇编](https://baike.baidu.com/item/%E6%B1%87%E7%BC%96" \t "_blank)语言程序设计第二版》。

1. **实验环境**

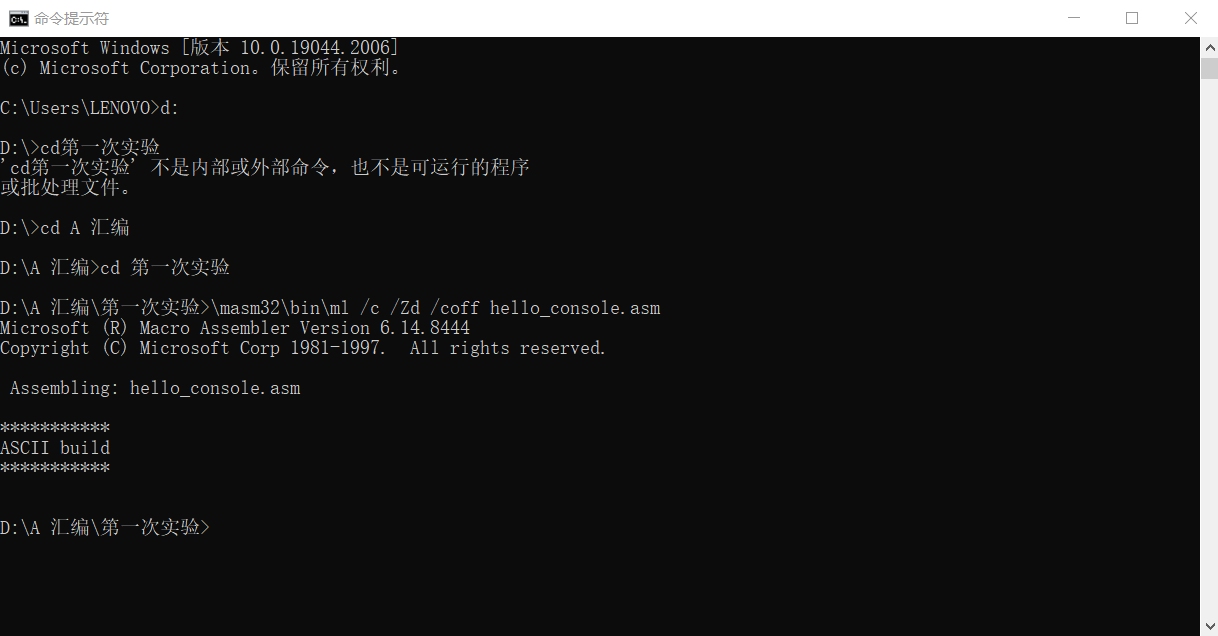
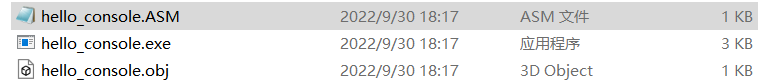
Windows操作系统，MASM32编译环境。

1. **实验步骤：命令行输出“HelloWorld”**
2. 源文件：用文本编辑器编写的asm文本文件

****

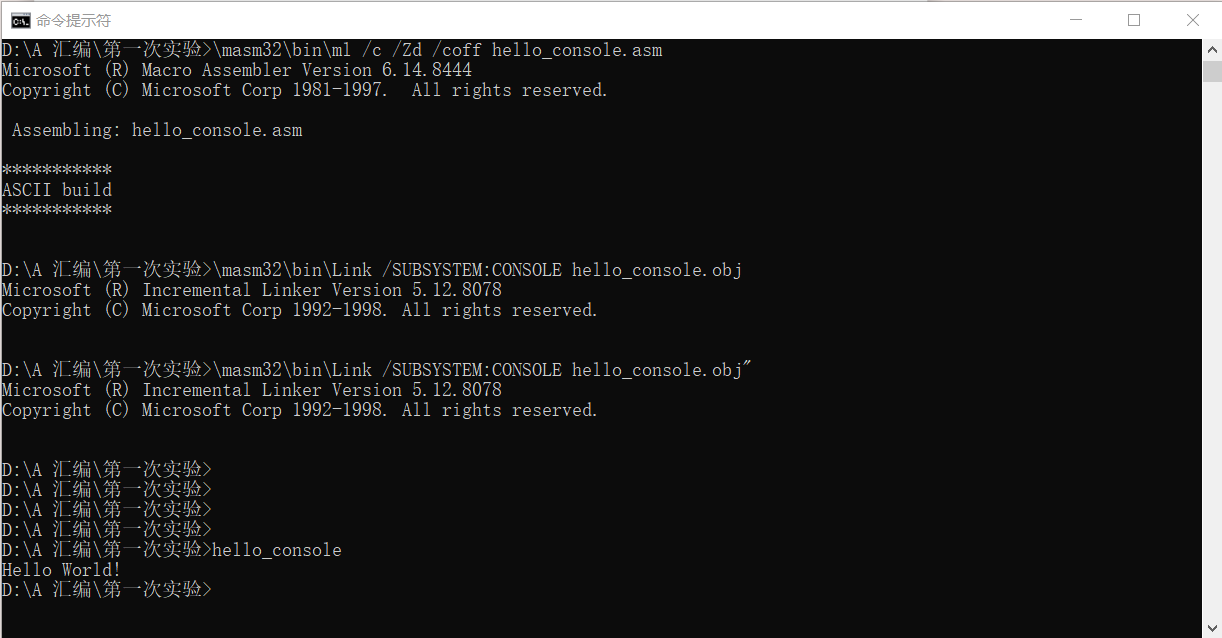
1. 汇编：用汇编程序（\masm32\bin\ml.exe）对源程序进行汇编，形成目标文件（.obj），格式如下：

“\masm32\bin\ml /c /Zd /coff hello\_console.asm”

****

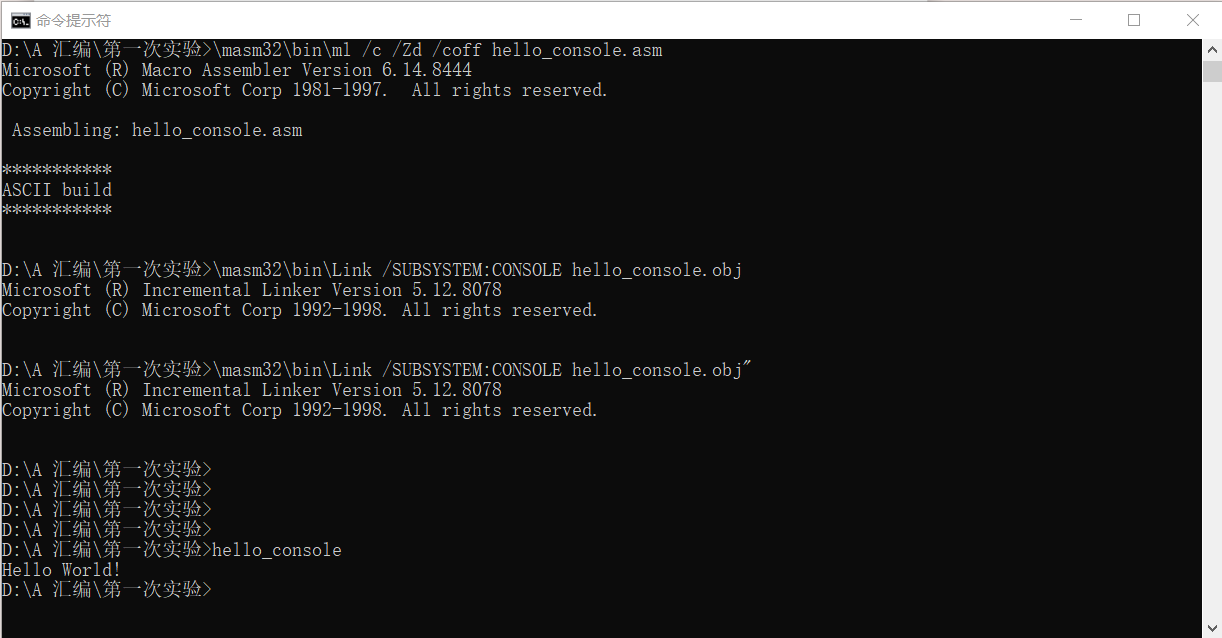
1. 连接：用连接程序（\masm32\bin\link.exe）对目标程序进行连接，形成可执行文件（.exe），格式如下

“\masm32\bin\Link /SUBSYSTEM:CONSOLE hello\_console.obj”

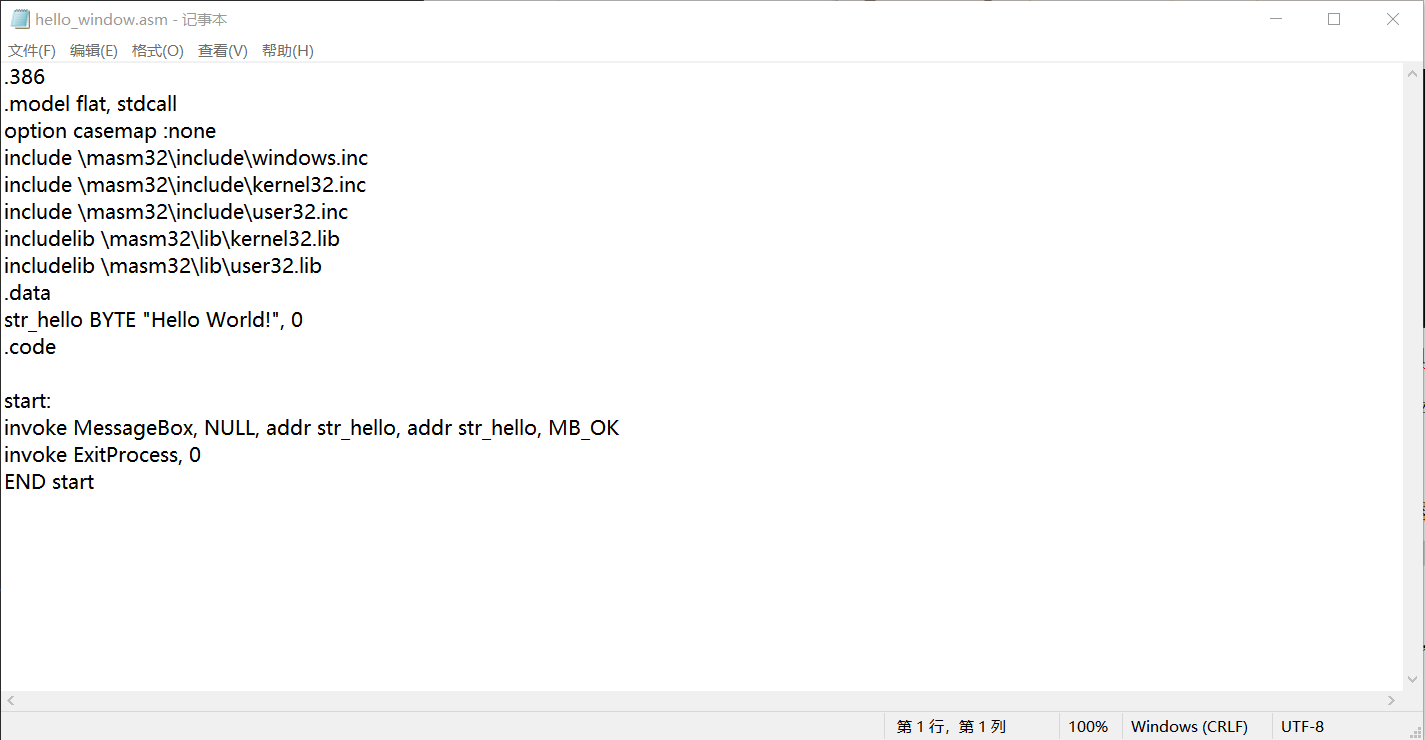


1. **实验命令行输出“HelloWorld”截图**

输入.asm文件名

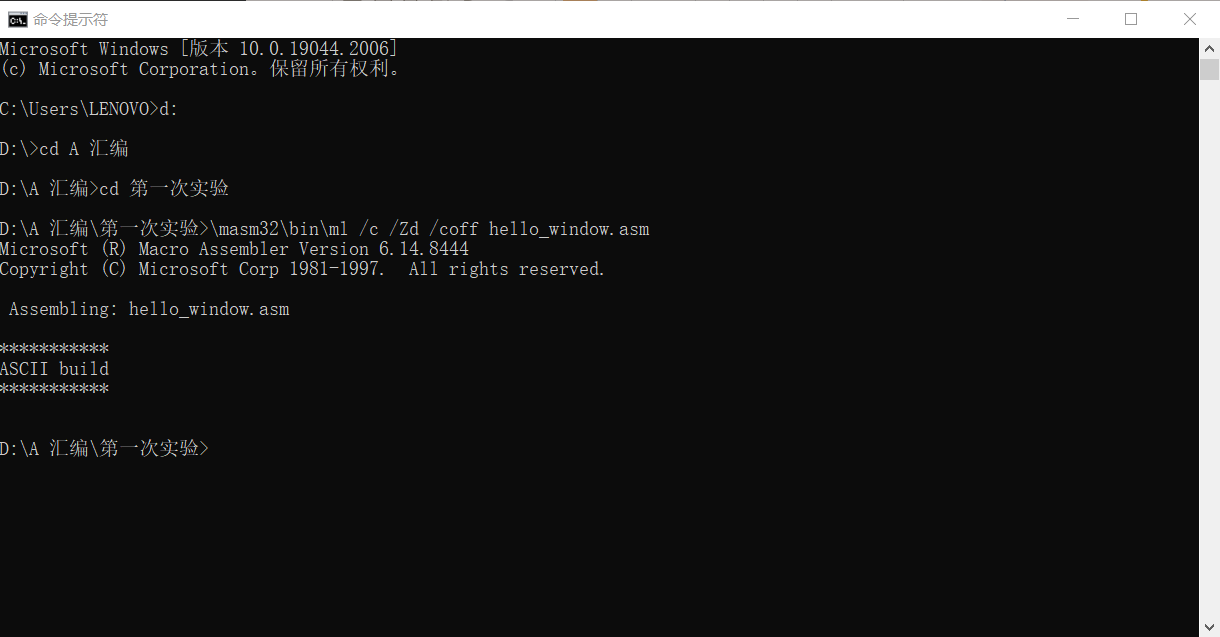


1. **实验步骤：窗口输出“HelloWorld”**
2. 源文件：用文本编辑器编写的asm文本文件

****

1. 汇编：用汇编程序（\masm32\bin\ml.exe）对源程序进行汇编，形成目标文件（.obj），格式如下：

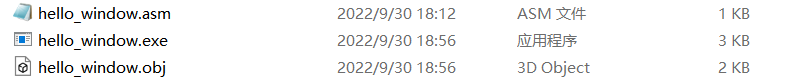
“\masm32\bin\ml /c /Zd /coff hello\_window.asm”

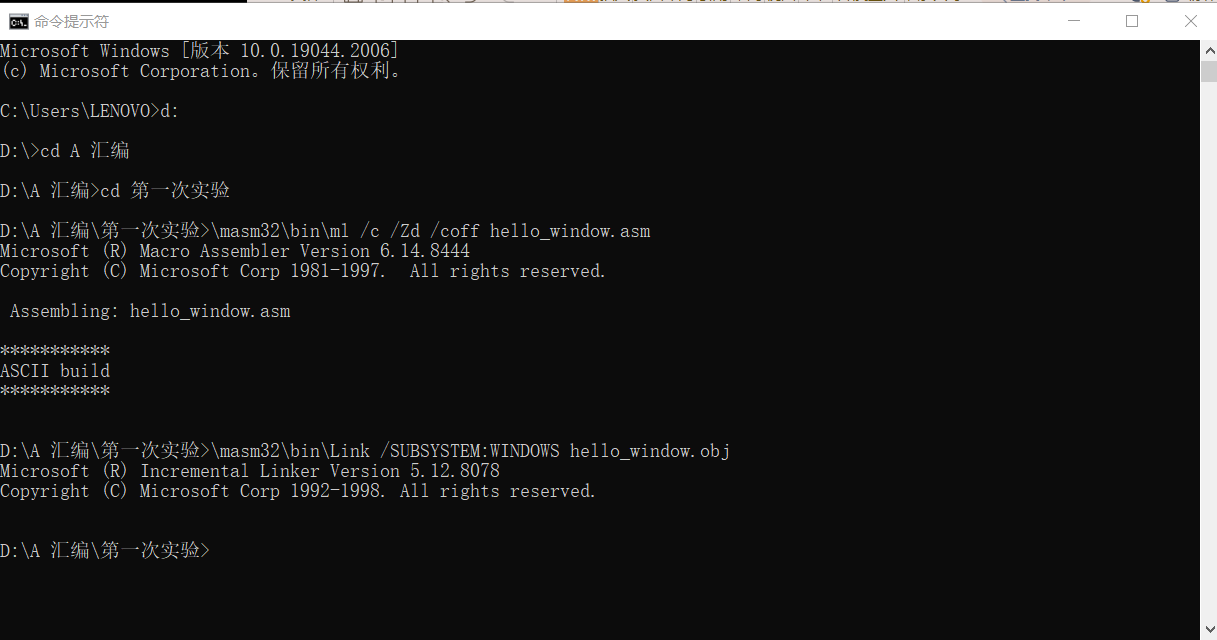
****

w2

1. 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），格式如下

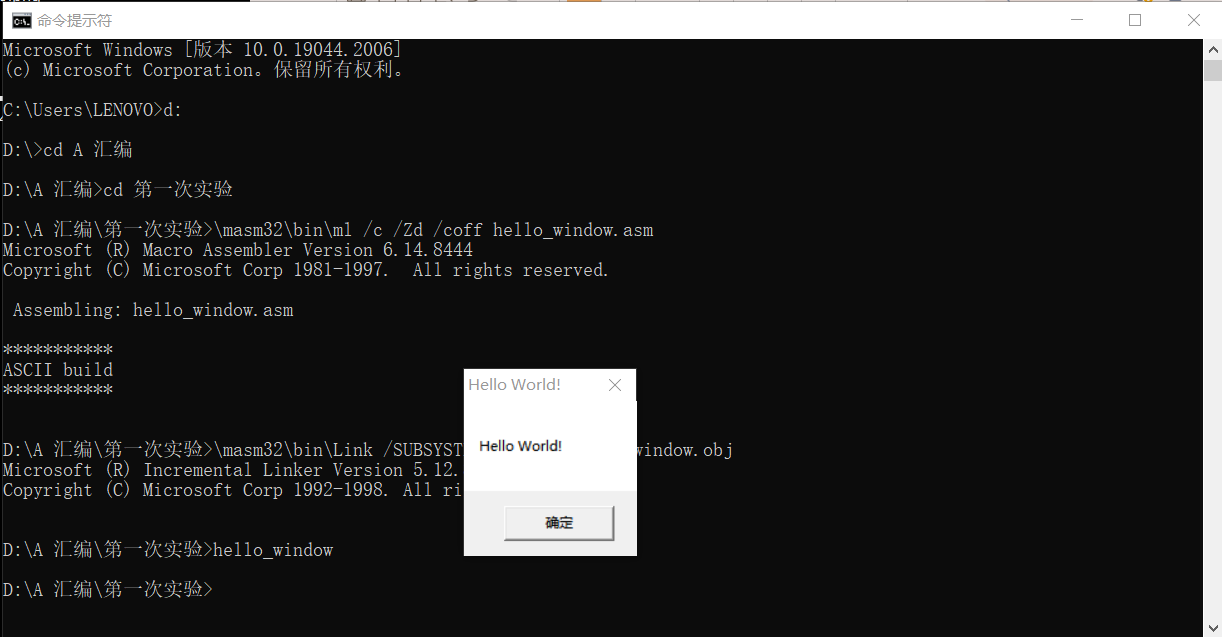
“\masm32\bin\Link /SUBSYSTEM:WINDOWS hello\_window.obj”





1. **实验窗口输出“HelloWorld”截图**

输入.asm文件名



1. **汇编命令与参数的解析**
2. “\masm32\bin\ml /c /Zd /coff hello\_console.asm”
3. \masm32\bin\ml /告诉要调用什么
4. ml 程序可以用来汇编并链接一个或多个汇编语言源文件
5. **ml /c /Zd /coff**

**/c 是告诉MASM只编译不连接。**

**/coff 告诉MASM**生成Microsoft公共目标文件格式的文件

**/Zd--Add line number debug info加上行号调试信息。**

1. hello\_console.asm 告诉执行文件在信息
2. “\masm32\bin\link /SUBSYSTEM:CONSOLE hello\_console.obj”
3. 告诉要使用连接，连接使用link命令 (masm32\bin\link.exe)
4. /SUBSYSTEM:选择运行环境（console命令行或WindowsGUI）
5. **SUBSYSTEM:CONSOLE** 生成命令行程序
6. hello\_console.obj 告诉执行文件信息
7. **汇编程序解析**

.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，.386兼容性相对最好）

.model flat, stdcall（flat：平坦模式，4GB内存空间，stdcall : 调用约定，stdcall是Win32 API函数的调用约定）

option casemap :none（不区分大小写）

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

include \masm32\include\kernel32.inc（windows一些函数在kernel库里）

include \masm32\include\masm32.inc（masm32也提供了些函数）

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

includelib \masm32\lib\masm32.lib（includelib …lib 是链接库）

.data（定义已初始化数据段的开始）

str\_hello BYTE "Hello World!", 0（0告诉字符串Hello World!已经结束到末尾了）

.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, 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（Kernel32.inc中定义的函数，退出程序执行）

END start（标记模块的结束，指定程序的入口点，告诉CPU从start开始执行，并把start写到寄存器里）

.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，.386兼容性相对最好）  
.model flat, stdcall （flat：平坦模式，4GB内存空间，stdcall : 调用约定，stdcall是Win32 API函数的调用约定）

option casemap :none （不区分大小写）

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

include \masm32\include\kernel32.inc （windows一些函数在kernel库里）

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib （链接kernel32的库）

includelib \masm32\lib\user32.lib （includelib …lib 链接库）

.data（定义已初始化数据段的开始）  
str\_hello BYTE "Hello World!", 0 （0告诉字符串Hello World!已经结束到末尾了）

.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（窗口输出 Hello World!）

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

END start（标记模块的结束，指定程序的入口点，告诉CPU从start开始执行，并把start写到寄存器里）