**实验1 HelloWorld**

**一、实验目的**

1.熟悉Win32汇编MASM32的编译环境；

2.命令行输出“HelloWorld”

3.窗口输出“HelloWorld”

**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，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)，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)[编译器](https://baike.baidu.com/item/%E7%BC%96%E8%AF%91%E5%99%A8)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)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》。

**三、实验环境**

Windows操作系统，MASM32编译环境。

**四、实验内容及程序解析**

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。

**汇编程序1——hello\_console.asm**

.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

// .model 初始化程序的内存模式

// 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

// 标签：大小写敏感

include \masm32\include\windows.inc

// 包含了Win32 API 的一些常量和函数定义

include \masm32\include\kernel32.inc

// 包含了后面使用的ExitProcess函数——ExitProcess函数是kernel32.inc中定义的函数，退出程序执行

include \masm32\include\masm32.inc

// 包含了后面使用的StdOut函数——StdOut函数是masm32.inc中定义的函数，将内存数据输出到命令行窗口上

includelib \masm32\lib\kernel32.lib

// 链接库

includelib \masm32\lib\masm32.lib

// 链接库

.data

// 定义已初始化数据段的开始

str\_hello BYTE "Hello World!", 0

// 定义字符串“Hello World！”且字符串的结尾是0

.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

// masm32.inc中定义的函数，将HelloWorld输出到命令行窗口上

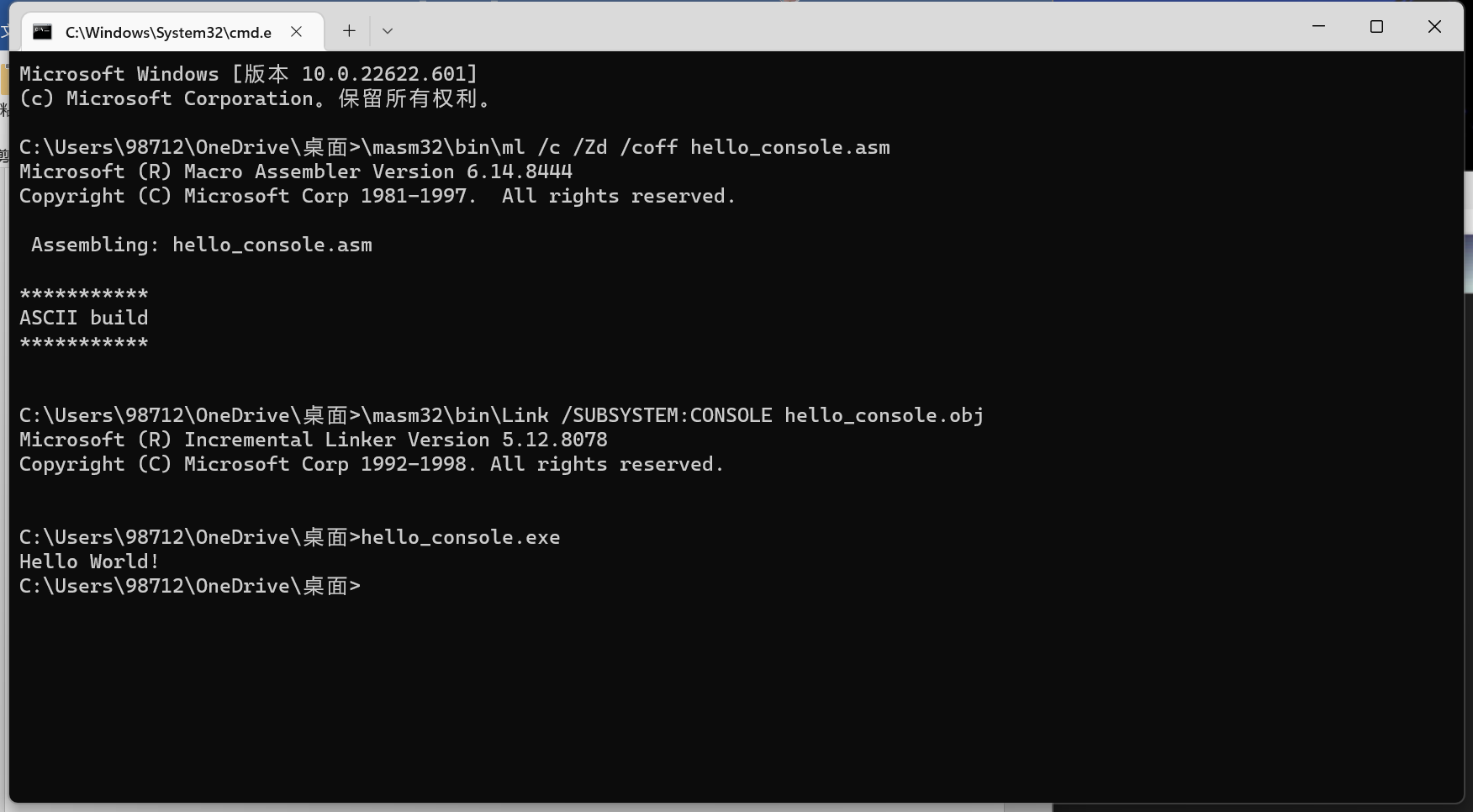
invoke ExitProcess, 0

// Kernel32.inc中定义的函数，退出程序执行

END start

//标记模块的结束或指定程序的入口点

**实验结果如下：**



**汇编程序2——hello\_window.asm**

.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

// .model 初始化程序的内存模式

// 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

// 标签：大小写敏感

include \masm32\include\windows.inc

// 包含了Win32 API 的一些常量和函数定义

include \masm32\include\kernel32.inc

// 包含了后面使用的ExitProcess函数——ExitProcess函数是kernel32.inc中定义的函数，退出程序执行

include \masm32\include\user32.inc

// 包含了后面使用的MessageBox函数——MessageBox函数是user32.inc中定义的函数，将内存数据输出到对话框上

includelib \masm32\lib\kernel32.lib

// 链接库

includelib \masm32\lib\user32.lib

// 链接库

.data

// 定义已初始化数据段的开始  
str\_hello BYTE "Hello World!", 0

// 定义字符串“Hello World！”且字符串的结尾是0

.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

// user32.inc中定义的函数，将HelloWorld输出到对话框上

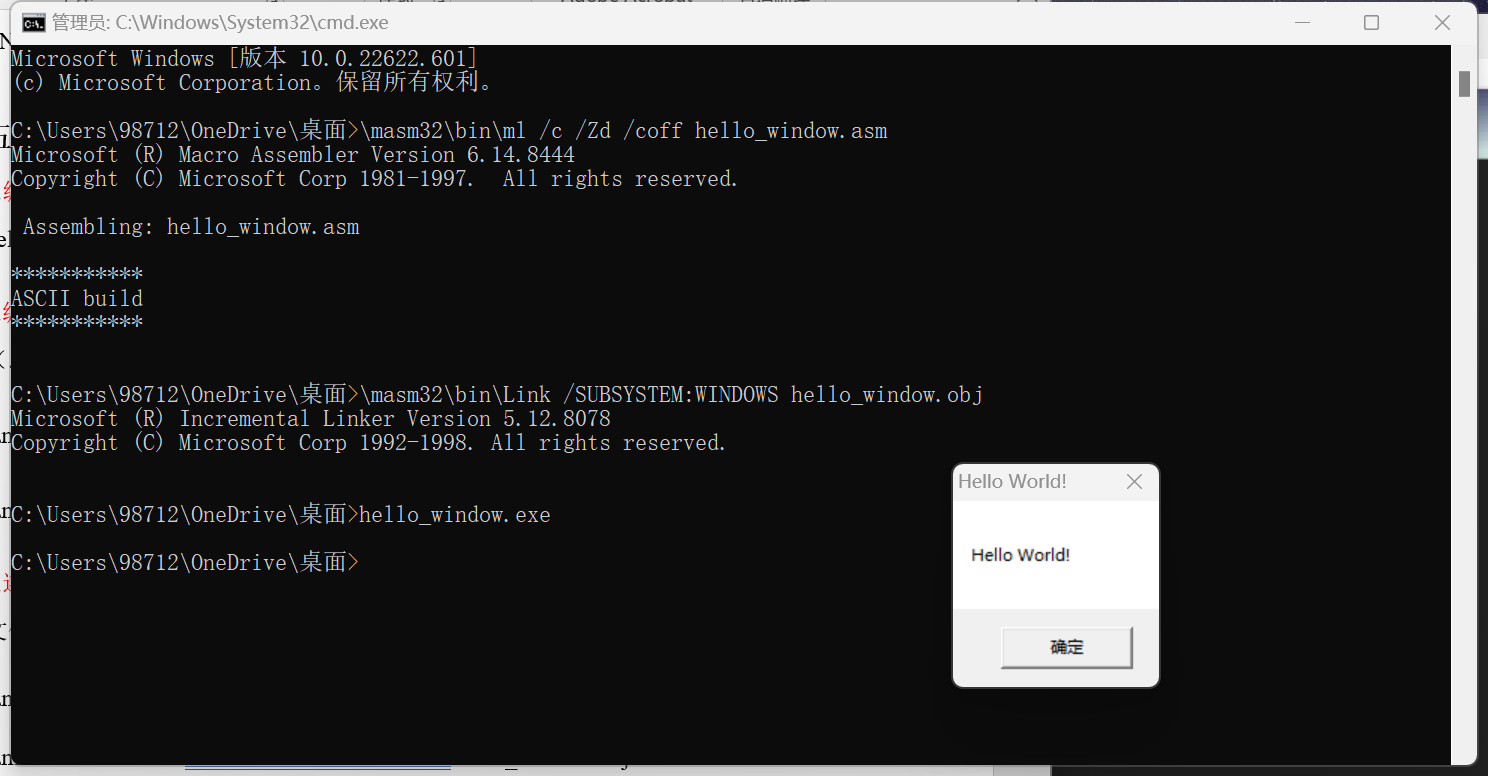
invoke ExitProcess, 0

// Kernel32.inc中定义的函数，退出程序执行

END start

//标记模块的结束或指定程序的入口点

**实验结果如下：**



**五、实验步骤**

1.编辑：用编辑软件（Notepad）形成源程序（.asm）,如：hello\_console.asm和hello\_window.asm.

2.编译：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”

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

3.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”

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

4.执行：如果结果在屏幕在显示，则直接执行可执行文件。

**六、汇编命令和参数的解析**

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

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

编译指令：汇编器把汇编源文件翻译成机器语言，生成目标文件。

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

• 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 只编译、不链接

• /Zd 在目标文件中生成行号信息

• /coff 生成Microsoft公共目标文件格式（common object file format）的文件

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

“\masm32\bin\link /SUBSYSTEM:CONSOLE hello\_window.obj”

连接指令：链接器从库中复制所需的过程，并将其同目标文件合并在一起生成可执行文件。

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）。

• Link.exe 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