**汇编语言与逆向技术实验报告**

**Lab1-HelloWorld**

**学号：2110957 姓名：蒋薇 专业： 计算机科学与技术**

1. **实验内容**

用文本编辑程序写程序，形成.ASM文件，用汇编程序对.ASM文件进行汇编，形成.OBJ文件，再用连接程序对.OBJ文件进行连接，形成.EXE文件。

熟悉Win32汇编MASM32的编译环境；

命令行输出“HelloWorld!”字符串的汇编程序，

窗口输出“HelloWorld!”字符串的汇编程序。

1. **实验步骤**

实验过程中的详细步骤，包括关键代码与截图等。

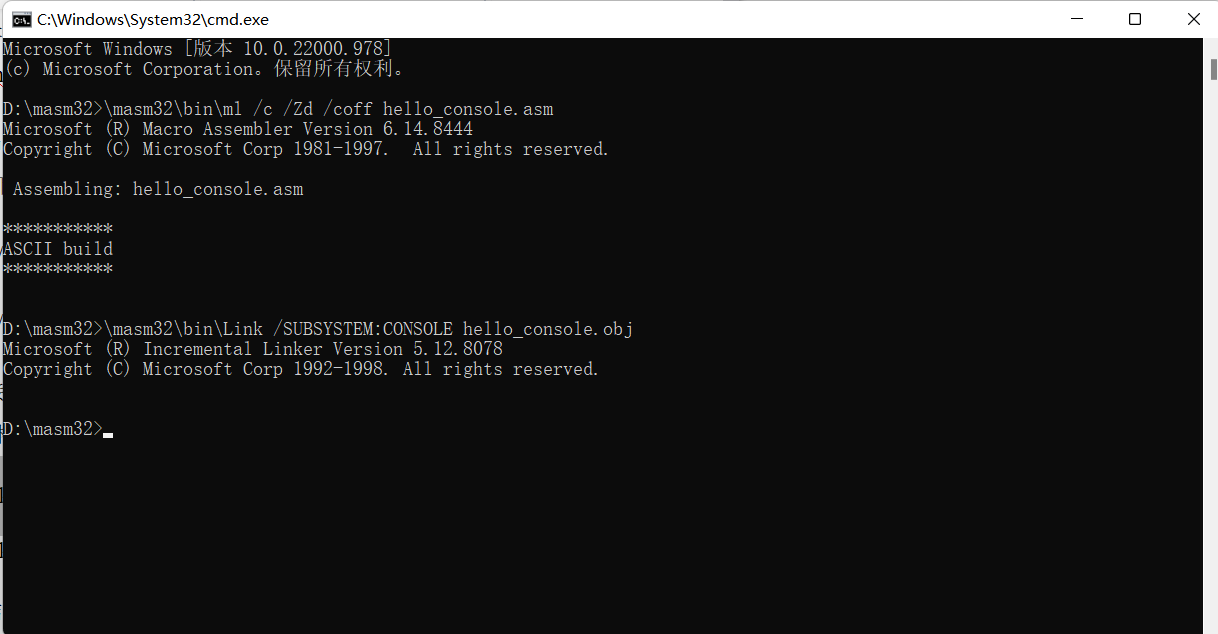
hello\_console.asm

1. 编辑 新建一个名为hello\_console.asm 的文件,用文本编辑器将源程序写入该文件中。



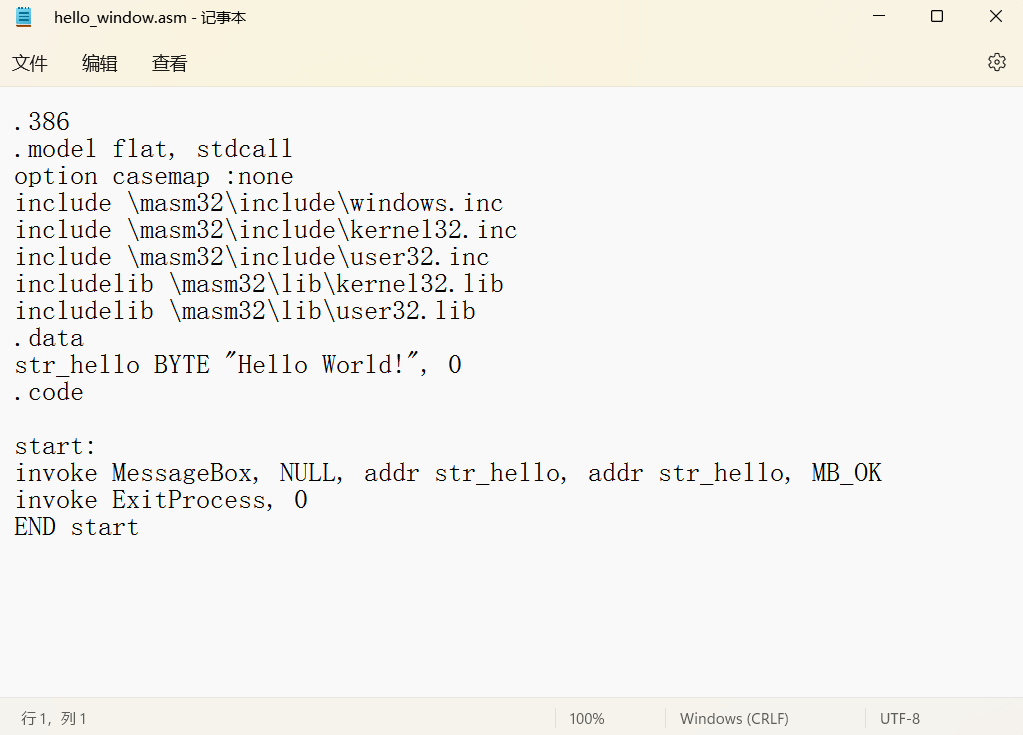
2. 汇编 汇编程序的作用是 把汇编语言源程序翻译成为机器代码，产生二进制格式的目标文件（Object File） 在DOS中输入以下命令： “\masm32\bin\ml /c /Zd /coff hello\_console.asm”

3. 连接 连接就是使用连接程序LINK 把目标文件（OBJ）转换为可执行的EXE文件 。 在DOS中输入以下命令：“\masm32\bin\Link /SUBSYSTEM:CONSOLE hello\_console.obj”



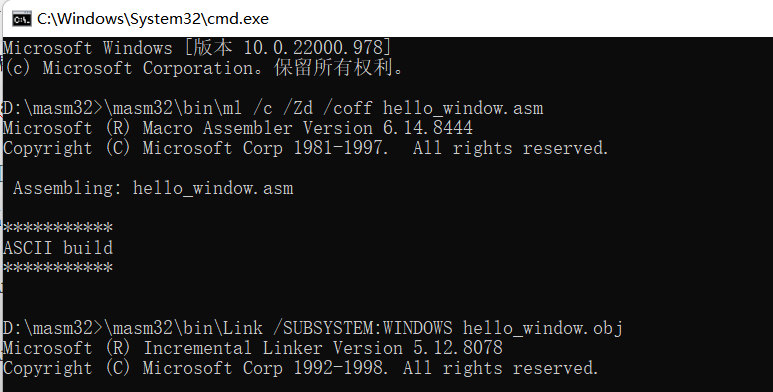
hello\_window.asm

1编辑 新建一个名为hello\_console.asm 的文件,用文本编辑器将源程序写入该文件中。

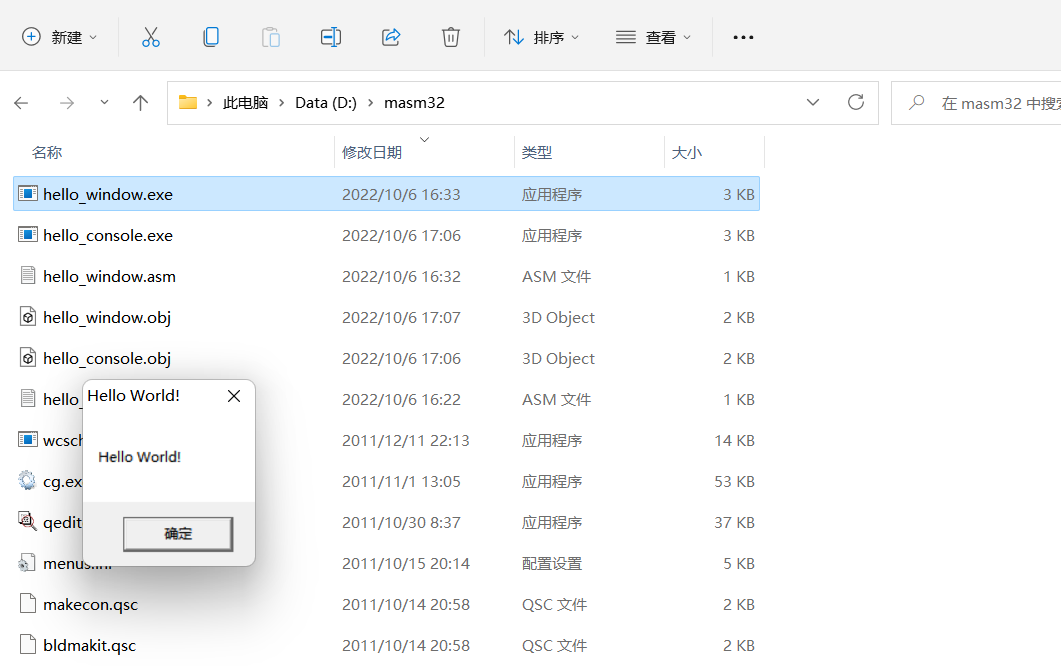


2. 汇编 汇编程序的作用是 把汇编语言源程序翻译成为机器代码，产生二进制格式的目标文件（Object File） 在DOS中输入以下命令： “\masm32\bin\ml /c /Zd /coff hello\_window.asm”

3. 连接 连接就是使用连接程序LINK 把目标文件（OBJ）转换为可执行的EXE文件 。 在DOS中输入以下命令：“\masm32\bin\Link /SUBSYSTEM:WINDOWS hello\_window.obj”



1. **实验截图**



1. **实验心得**
2. 汇编命令和参数的解析：

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

编译

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

链接

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

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

1. 汇编程序解析

每一行汇编语句的含义

.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//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//大小写敏感

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\masm32.inc// 函数的常量和声明

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\masm32.lib// 链接库

//定义已初始化数据段的开始

.data

str\_hello BYTE "Hello World!", 0

//定义代码段的开始

.code

//指令标号，标记指令地址

start:

//StdOut, masm32.inc中定义的函数，将内存数据输出到命令行窗口上

invoke StdOut, addr str\_hello

//ExitProcess， Kernel32.inc中定义的函数，退出 程序执行

invoke ExitProcess, 0

//标记模块的结束

//指定程序的入口点

END start