

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，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。

MASM 的 windows 编程的教学书籍有《windows 环境下 32 位汇编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

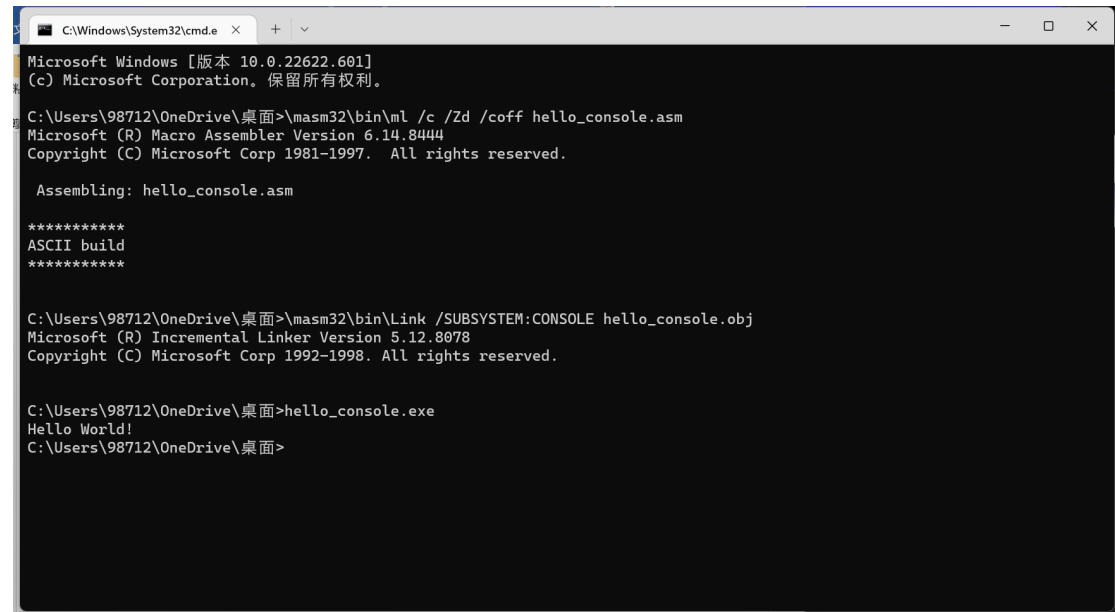
invoke ExitProcess, 0

// 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：



```
C:\Windows\System32\cmd.e  X  +  -  X
Microsoft Windows [版本 10.0.22622.601]
(c) Microsoft Corporation。保留所有权利。

C:\Users\98712\OneDrive\桌面>masm32\bin\ml /c /Zd /coff hello_console.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

    Assembling: hello_console.asm

*****
ASCII build
*****

C:\Users\98712\OneDrive\桌面>masm32\bin\Link /SUBSYSTEM:CONSOLE hello_console.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

C:\Users\98712\OneDrive\桌面>hello_console.exe
Hello World!
C:\Users\98712\OneDrive\桌面>
```

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

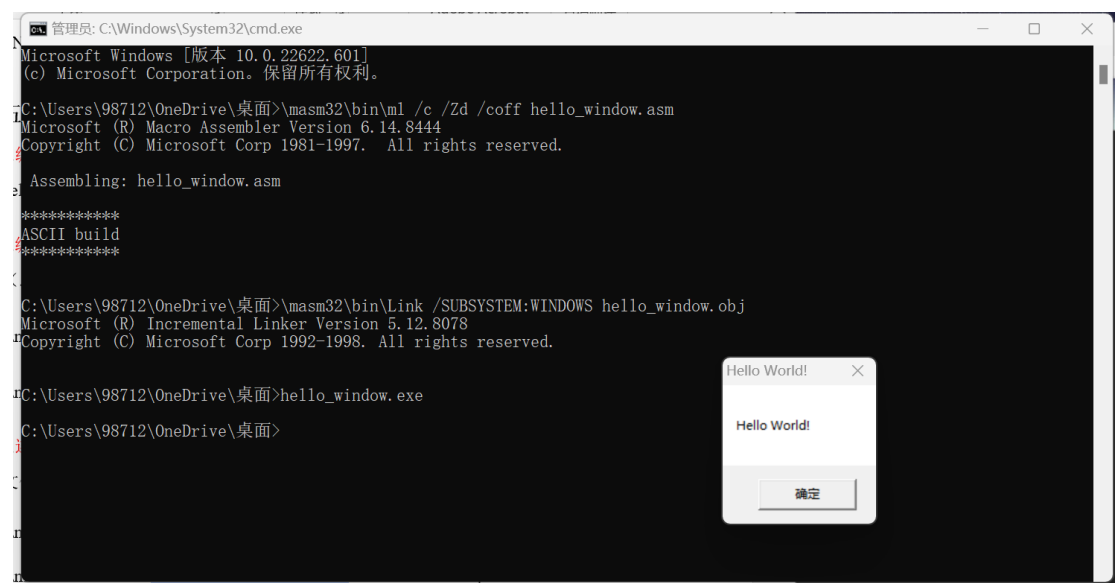
// 链接库

```

.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