

汇编语言与逆向技术实验报告

Lab1-HelloWorld

学号：2112514 姓名：辛浩然 专业：信息安全、法学

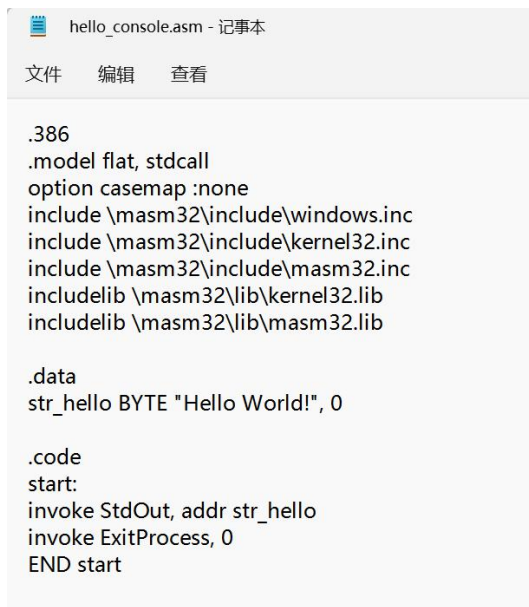
一、实验内容

实现一个在命令行输出“HelloWorld”字符串的汇编程序，和一个在 Windows MessageBox 中输出“HelloWorld”的汇编程序。

二、实验步骤

1.在命令行输出“HelloWorld”字符串的汇编程序：

(1) 编辑：形成源程序：



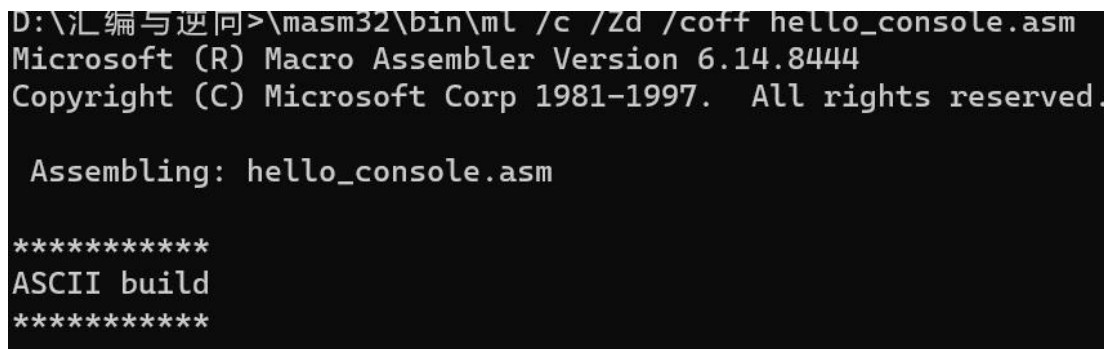
```
hello_console.asm - 记事本
文件 编辑 查看

.386
.model flat, stdcall
option casemap :none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
include \masm32\include\masm32.inc
includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\masm32.lib

.data
str_hello BYTE "Hello World!", 0

.code
start:
invoke StdOut, addr str_hello
invoke ExitProcess, 0
END start
```

(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）：



```
D:\汇编与逆向>\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
*****
```

(3) 链接：用链接程序（\masm32\bin\link.exe）对目标程序进行链接，

形成可执行文件（.exe）：

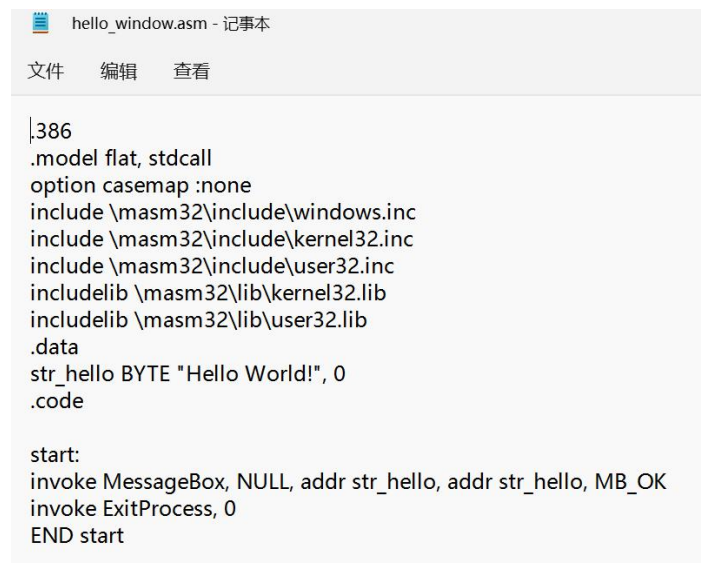
```
D:\汇编与逆向>\masm32\bin\link /SUBSYSTEM:CONSOLE hello_console.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

（4）执行：

```
D:\汇编与逆向>.\hello_console.exe
Hello World!
```

2.在 Windows MessageBox 中输出“HelloWorld”的汇编程序：

（1）编辑：形成源程序：



```
hello_window.asm - 记事本
文件 编辑 查看

|386
.model flat, stdcall
option casemap:none
include \masm32\include\windows.inc
include \masm32\include\kernel32.inc
include \masm32\include\user32.inc
includelib \masm32\lib\kernel32.lib
includelib \masm32\lib\user32.lib
.data
str_hello BYTE "Hello World!", 0
.code

start:
invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK
invoke ExitProcess, 0
END start
```

（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）：

```
D:\汇编与逆向>\masm32\bin\ml /c /Zd /coff hello_window.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
```

```
Assembling: hello_window.asm
```

```
*****
ASCII build
*****
```

（3）链接：用链接程序（\masm32\bin\link.exe）对目标程序进行链接，形成可执行文件（.exe）：

```
D:\汇编与逆向>\masm32\bin\Link /SUBSYSTEM:WINDOWS hello_window.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

(4) 执行:



三、 实验截图

在命令行输出“HelloWorld”字符串

```
D:\汇编与逆向>\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
*****

D:\汇编与逆向>\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.

D:\汇编与逆向>.\hello_console.exe
Hello World!
```

在 Windows MessageBox 中输出“HelloWorld”

```
D:\汇编与逆向>\masm32\bin\ml /c /Zd /coff hello_window.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: hello_window.asm

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

D:\汇编与逆向>\masm32\bin\Link /SUBSYSTEM:WINDOWS hello_window.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

D:\汇编与逆向>.\hello_window.exe
```



四、 实验分析

1. 汇编命令和参数的解析:

(1) 命令行输出实验:

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

用汇编程序 (\masm32\bin\ml.exe) 对 hello_console.asm 进行汇编, 形成目标文件 (.obj)。其中: /c 是只汇编、不链接的指令, /Zd 是在目标文件中生成行号信息, 即目标文件指令与源代码中代码行的对应关系, /coff 是生成 microsoft 公共目标文件格式的文件。

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

用链接程序 (\masm32\bin\link.exe) 对 hello_console.obj 进行链接, 形成可执行文件 (.exe)。/SUBSYSTEM:CONSOLE 是生成命令行程序的指令。

(2) 窗口输出实验:

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

对 hello_window.asm 进行汇编, 汇编生成 hello_window.obj 文件。其中: /c 是只汇编、不链接的指令, /Zd 是在目标文件中生成行号信息, 即目标文件指令与源代码中代码行的对应关系, /coff 是生成 microsoft 公共目标文件格式的文件。

② “\masm32\bin\Link /SUBSYSTEM:WINDOWS hello_window.obj”

用链接程序 (\masm32\bin\link.exe) 对 hello_window.obj 进行链接, 形成可执行文件 (.exe)。/SUBSYSTEM:WINDOWS 是生成窗口程序的指令。

2. 汇编程序解析

(1) hello_console.asm

.386

; 表示程序使用的指令集, 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, 使用平坦内存模式 (4GB 内存空间) 并使用 stdcall 调用习惯, 即 API 调用时右边的参数先入栈。

option casemap :none

; 编译器程序中变量名和子程序名对大小写敏感。

include \masm32\include\windows.inc

```
include \masm32\include\kernel32.inc
```

```
include \masm32\include\masm32.inc
```

; include 跟在其后的文件名所指定的文件在编译时将插入在该处。这三条语句得到函数的常量和声明。

```
includelib \masm32\lib\kernel32.lib
```

```
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
```

; byte 定义字符串，命名为 str_hello，0 表示字符串的结尾。

```
.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 为 masm32.inc 中定义的函数，将内存数据输出到命令行窗口。

addr 用来把标号的地址传递给被调用的函数。

```
invoke ExitProcess, 0
```

; ExitProcess 为 kernel32.inc 中定义的函数，退出程序执行

```
END start ; 标记模块的结束，指定程序的入口点是 start
```

(2) hello_window.asm

大部分语句与上一程序解析相同

```
.386
```

```
.model flat, stdcall
```

```
option casemap :none
```

```
include \masm32\include\windows.inc
```

```
include \masm32\include\kernel32.inc
```

```
include \masm32\include\user32.inc
```

```
includelib \masm32\lib\kernel32.lib
```

```
includelib \masm32\lib\user32.lib
```

```
.data
```

```
str_hello BYTE "Hello World!", 0
```

```
.code
```

```
start:
```

```
invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK
```

; 调用 MessageBox，用来弹出一个对话框，标题和显示的内容均为 str_hello，包含一个确定按钮。

```
invoke ExitProcess, 0
```

```
END start
```