

# 南开大学

## 汇编语言与逆向技术课程实验报告

### 实验一：Hello World



学 院 网络空间安全学院  
专 业 信息安全、法学双学位  
学 号 2212000  
姓 名 宋奕纬  
班 级 1061

## 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” ；
- 4、简单掌握一定汇编语言的知识。

## 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。

## 三、实验过程

- 1、编辑：用记事本形成两个源程序 hello\_console.asm 和 hello\_window.asm.

\*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
```

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
```

名称	修改日期	类型
 hello_window.asm	2023/10/8 10:31	Assembler Source
 hello_console.asm	2023/10/8 10:29	Assembler Source

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）

在命令行中输入如下指令对源程序进行汇编

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.22621.2283]
(c) Microsoft Corporation。保留所有权利。

C:\Users\宋奕纬>D:

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

Assembling: D:\hb\hello_console.asm



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

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

Assembling: D:\hb\hello_window.asm

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



```

 hello_console.obj	2023/10/8 21:49	Object File	2 KB
 hello_window.obj	2023/10/8 21:49	Object File	2 KB

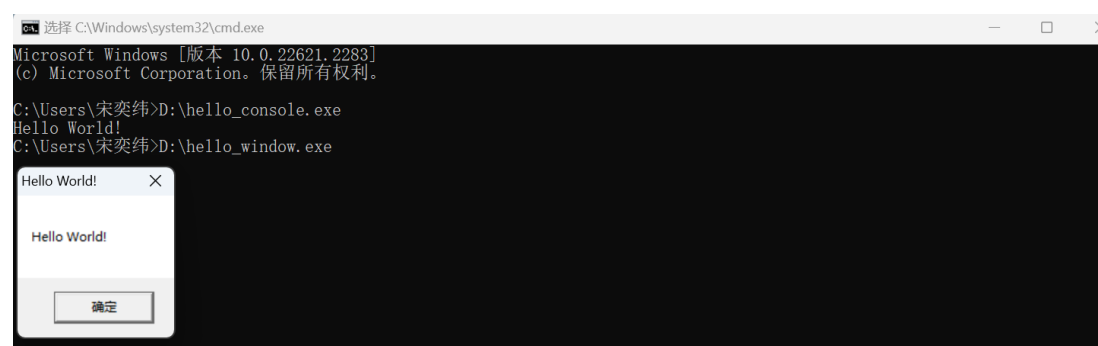
3、连接：用连接程序（\masm32\bin\link.exe）对目标程序进行连接，形成可执行文件（.exe）

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

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

 hello_console.exe	2023/10/8 21:57	应用程序	3 KB
 hello_window.exe	2023/10/8 21:58	应用程序	3 KB

4、执行：如果结果在屏幕在显示，则直接执行可执行文件。



## 四、实验结论及心得体会

1、汇编命令和参数的解析：

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

①\masm32\bin\ml：表示使用 masm32 程序集里的汇编编译器 ml.exe。ml.exe 可以将汇编代码转化为一个目标文件（此处为.obj）。

②/c：表示只编译源代码，不进行连接操作，输出的是目标文件（此处为.obj）。

③/Zd：产生调试信息，生成一个带有调试信息的.obj 文件。

④/coff：生成的目标文件格式为 COFF 格式，这个生成的文件可以被链接器（如 link.exe）所识别。

⑤hello\_console.asm 代表需要编译的源代码文件名。

该命令将会把 `hello_console.asm` 编译为 `hello_console.obj` 文件，`.obj` 文件可被进一步用于链接生成可执行文件。

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

① `\masm32\bin\link` 表示使用 `masm32` 程序集里的汇编编译器 `link.exe`。  
`link.exe` 可以将汇编生成的 `obj` 文件和其他静态库或 `DLL` 文件链接成一个可执行文件（`.exe`）

② `/SUBSYSTEM:CONSOLE`：指定生成的可执行文件使用控制台窗口方式启动。即在启动可执行程序时打开一个命令行窗口用于程序的输出。

③ `hello_console.obj` 代表需要进行链接的目标文件名。

这条命令执行后将会把 `hello_console.obj` 与其所依赖的库文件连接起来，最终生成一个名为 `hello_console.exe` 的可执行文件。

## 2、汇编程序解析

(1) 汇编程序 1: `hello_console.asm`

<code>.386</code>	指定使用 80386 处理器及以上版本的指令集。
<code>.model flat, stdcall</code>	指定内存模型为 <code>flat</code> ，调用约定为 <code>stdcall</code> 。内存模型 <code>flat</code> 表示所有数据和代码在单一平坦的地址空间内， <code>stdcall</code> 是一种函数调用约定，它规定函数参数从右往左压栈，调用方负责清理堆栈上的参数。
<code>option casemap :none</code>	指定不对标识符进行大小写转换。
<code>include \masm32\include\windows.inc include \masm32\include\kernel32.inc include</code>	<code>include</code> 引入需要用到的头文件，其中包括 <code>windows.inc</code> 、 <code>kernel32.inc</code> 、 <code>masm32.inc</code> 三

<code>\masm32\include\masm32.inc</code>	个头文件。
<code>includelib \masm32\lib\kernel32.lib includelib \masm32\lib\masm32.lib</code>	<code>includelib</code> 用于链接静态库文件。指定 <code>kernel32.lib</code> 和 <code>masm32.lib</code> 程序，在链接阶段引入这些库中的函数。
<code>.data</code>	定义数据段。在数据段中，可以声明和初始化全局变量和字符串等数据。
<code>str_hello BYTE "Hello World!", 0</code>	在数据段中定义一个名为 <code>str_hello</code> 的字符串变量，它使用 <code>BYTE</code> 指令表示以字节方式存储。该字符串的内容是 "Hello World!"，并以 0 作为结束符
<code>.code</code>	定义代码段。在代码段中，程序可以编写实际的汇编代码。
<code>start:</code>	程序的入口点，标记了程序执行的起始位置。
<code>invoke StdOut, addr str_hello invoke ExitProcess, 0</code>	<code>Invoke</code> 是宏定义，用于调用函数。此处调用 <code>StdOut</code> 函数，将 <code>str_hello</code> 字符串输出到控制台窗口中，然后调用 <code>ExitProcess</code> 函数主动退出程序并返回 0。
<code>END start</code>	表示代码块结束。

## (2) 汇编程序 2: `hello_window.asm`

<code>.386</code>	指定使用 80386 处理器及以上版本的指令集。
<code>.model flat, stdcall</code>	指定内存模型为 <code>flat</code> ，调用约定为 <code>stdcall</code> 。内存模型 <code>flat</code> 表示所有数据和代码在单一平坦的地

	址空间内,stdcall 是一种函数调用约定,它规定函数参数从右往左压栈,调用方负责清理堆栈上的参数。
option casemap :none	指定不对标识符进行大小写转换。
include \masm32\include\windows.inc include \masm32\include\kernel32.inc include \masm32\include\user32.inc	include 引入需要用到的头文件,其中包括 windows.inc、kernel32.inc、user32.inc 三个头文件。
includelib \masm32\lib\kernel32.lib includelib \masm32\lib\user32.lib	includelib 用于链接静态库文件。指定 kernel32.lib 和 user32.lib 程序,在链接阶段引入这些库中的函数。
.data	定义数据段。在数据段中,可以声明和初始化全局变量和字符串等数据。
str_hello BYTE "Hello World!", 0	在数据段中定义一个名为 str_hello 的字符串变量,它使用 BYTE 指令表示以字节方式存储。该字符串的内容是 "Hello World!",并以 0 作为结束符
.code	定义代码段。在代码段中,程序可以编写实际的汇编代码。
start:	程序的入口点,标记了程序执行的起始位置。



<pre> invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK  invoke ExitProcess, 0 </pre>	<p>Invoke 是宏定义,用于调用函数。此处调用 MessageBox 函数,生成一个等待用户响应的消息窗,将 str_hello 字符串输出到标题栏文本为 str_hello 的消息框窗中,MB_OK 是一种消息框风格,指示将只显示一个“确定”按钮。然后调用 ExitProcess 函数主动退出程序并返回 0。</p>
<pre> END start </pre>	<p>表示代码块结束。</p>

### 3、收获与体会

- (1) 熟悉 Win32 汇编 MASM32 的编译环境;学会了利用汇编语言实现命令行输出和窗口输出“HelloWorld”。
- (2) 初步感受到汇编语言的结构与写法,感受到汇编之美与其中的乐趣。