南開大學

汇编语言与逆向技术课程实验报告

实验一：HelloWorld

学　　院　网络空间安全学院

专　　业　信息安全

学　　号　2313546

姓　　名　蒋枘言

班　　级　信息安全班

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、实验环境

Windows 操作系统，MASM32 编译环境。

三、实验原理

1.MASM32

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。

四、实验过程

1.编辑：用编辑软件（记事本）形成源程序 hello_console.asm 和 hello_window.asm。

●代码解析

（1）hello_console.asm

```
.386
; 指出该程序要求的最低 CPU（Intel 386）
.model flat, stdcall
; .model 用于初始化程序的内存模式
; flat 设置内存模型为平坦模型
; 并使用 stdcall 调用约定，stdcall 是 Win 32 API 的调用约定
option casemap :none
; 指定汇编器不区分大小写（大小写不敏感）
include \masm32\include\windows.inc
; 包含 Windows API 的定义和常量
include \masm32\include\kernel32.inc
; 包含 kernel32.dll 中函数的声明
include \masm32\include\masm32.inc
; 包含 MASM32 框架的声明和宏
includelib \masm32\lib\kernel32.lib
; 链接 kernel32.lib 库，提供 Windows API 的实现
includelib \masm32\lib\masm32.lib
; 链接 MASM32 库，提供框架功能
.data
; 定义已初始化数据段的开始
str_hello BYTE "Hello World!", 0
; 声明一个以 NULL 结尾的字符串常量
.code
; 定义代码段的开始
start:
```

```
; 程序入口点的标签
invoke StdOut, addr str_hello
; 调用 StdOut 宏（由 MASM32 框架提供），输出 str_hello 字符串
; StdOut 是 masm32.inc 中定义的函数，作用是将内存数据输出到命令行窗口上
; addr 操作符获取 str_hello 的地址
invoke ExitProcess, 0
; 调用 ExitProcess 函数，这表示程序正常退出
; ExitProcess 是 Kernel32.inc 中定义的函数，作用是退出程序执行
END start
; 指示汇编器程序的结束，并指定入口点为 start 标签
```

（2）hello_window.asm

```
.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 函数显示一个消息框
; MessageBox 的参数依次为：（下一行）
; 窗口句柄（NULL 表示无父窗口），消息文本，消息框标题，消息框选项
; addr str_hello 用于获取 str_hello 字符串的地址
; MB_OK 是消息框选项，表示只有一个“确定”按钮
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），格式如下：

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

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

●\masm32\bin\ml：这是 MASM32 汇编器的完整路径。它告诉操作系统在哪里找到 ml.exe 程序以执行汇编任务。\masm32\bin\ 是 MASM32 安装目录中包含 ml.exe 的文件夹。

●/c：这个选项指示汇编器仅进行编译，不进行链接。也就是说，它会生成目标文件（.obj），但不会尝试创建可执行文件（.exe）。

●/Zd：这个选项用于生成调试信息，并包含源代码行号。它使得生成的目标文件可以用于调试器，以便在调试时能够关联源代码行和生成的机器代码。

●/coff：这个选项指示汇编器生成 COFF（Common Object File Format）格式的目标文件。COFF 是一种用于存储程序对象代码的文件格式，它支持调试信息、重定位信息等。这是 Windows 平台上常用的目标文件格式之一。

●hello_console.asm 与 hello_window.asm：这是要编译的汇编源代码文件的名称。它应该位于当前工作目录或指定的路径中，以便汇编器可以找到并读取它。

3.链接：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:CONSOLE hello_window.obj”

●\masm32\bin\Link：这是链接器的完整路径，它告诉操作系统在哪里找到 link.exe 程序以执行链接任务。

●/SUBSYSTEM:CONSOLE：生成命令行程序。这个选项指定了生成的可执行文件的子系统类型。CONSOLE 表明生成的是一个控制台应用程序，这意味着当程序运行时，它会打开一个命令行窗口（如果尚未打开），并且所有的输入和输出都会通过这个窗口进行。

●hello_console.obj 和 hello_window.obj：这是要链接的目标文件的名称。它是之前通过汇编器编译 hello_console.asm 和 hello_window.asm 文件生成的。链接器会读取这个文件，并将其中的代码与其他必要的库代码和资源链接在一起，以生成最终的可执行文件。

4.执行：如果结果在屏幕在显示，则直接执行可执行文件。