汇编语言与逆向技术实验报告

Lab1-HelloWorld

学号：1911406

姓名：丁彦添

汇编命令和参数的解析

```
.\masm32\bin\ml /c /Zd /coff hello_console.asm
```

对于此编译命令，"\masm32\bin\ml" 表示使用编译器的地址。

"/c" 是告诉MASM只编译不链接。这主要是考虑到在链接前您可能还有其他工作要做。
"/coff" 告诉MASM产生的目标文件用 coff 格式。
"/Zd"加上行号调试信息。
当成功的编译了 hello_console.asm 后，编译器会产生 hello_console.obj 目标文件，目标文件中包含了以二进制形式存在的指令和数据，比可执行文件相差的只是链接器加入的重定位信息。

```
.\masm32\bin\link /SUBSYSTEM:CONSOLE hello_console.obj
```

/SUBSYSTEM:CONSOLE 告诉链接器可执行文件的运行平台是 CONSOLE 即命令行(另一个程序使用的平台是WINDOWS，即窗口对话框) 。

汇编程序解析

本次实验完成两个小型汇编程序，一个在 Windows 命令行输出 "Hello World!"，另一个在打开 Windows 会话窗输出 "Hello World!"。

具体汇编代码及每句汇编代码含义如下

```
.386--指定指令集，其中386比较老，跨平台兼容性比较好
.model flat, stdcall--指明子系统，调用约定。指定程序的内存模式为flat,在Win32下，只有一种内存模型，那就是FLAT。 stdcall告诉编译器参数的传递约定。参数的传递约定是指参数传递时的顺序(从左到右或从右到左)和由谁恢复堆栈指针(调用者或被调用者)。stdcall调用约定（定义如何传参等）。
option casemap :none--大小写约定，指明大小写是否敏感。此处强迫标签为字母大小写敏感。
-----------以下内容为包含的头文件及对应的库文件------------
include .\masm32\include\windows.inc--windows.inc 它包含了Win32API常数和定义的声明。
include .\masm32\include\kernel32.inc--kernel32.inc 它包含了退出函数。
include .\masm32\include\masm32.inc--masm32.inc 它包含了输出函数（显示在屏幕上）。它在masm32中内置着。
includelib .\masm32\lib\kernel32.lib
includelib .\masm32\lib\masm32.lib
--退出和输出函数须要有库libraries。
-------------------------------------------------------

.data--数据段，其内存属性是可读的
    str_hello BYTE "Hello World!", 0//定义一个字符串变量并且赋值

.code--代码段
start://告诉编译器程序的第一条指令从何开始，编译后会写到PE文件的入口点
```

```
    invoke StdOut, addr str_hello--调用标准输出StdOut，输出地址为str_hello的内容
    invoke ExitProcess, 0--调用结束进程的函数，退出进程
END start--结束
```

```
.386--指定指令集，其中386比较老，跨平台兼容性比较好
.model flat, stdcall--指明子系统，调用约定。指定程序的内存模式为flat,在Win32下，只有一种内存模型，那就是FLAT。 stdcall告诉编译器参数的传递约定。参数的传递约定是指参数传递时的顺序(从左到右或从右到左)和由谁恢复堆栈指针(调用者或被调用者)。stdcall调用约定（定义如何传参等）。
option casemap :none--大小写约定，指明大小写是否敏感。此处强迫标签为字母大小写敏感。
-----------以下内容为包含的头文件及对应的库文件------------
include .\masm32\include\windows.inc--windows.inc 它包含了Win32API常数和定义的声明。
include .\masm32\include\kernel32.inc--kernel32.inc 它包含了退出函数。
include .\masm32\include\user32.inc--user32.inc 它包含了输出函数（显示在对话框窗口上）。
includelib .\masm32\lib\kernel32.lib
includelib .\masm32\lib\user32.lib
--退出和输出函数须要有库libraries。
-------------------------------------------------------
.data--数据段，其内存属性是可读的
    str_hello BYTE "Servus!", 0--定义一个字符串变量并且赋值
    str_heading BYTE "Guten Tag!", 0

.code--代码段
start:--告诉编译器程序的第一条指令从何开始，编译后会写到PE文件的入口点
    invoke MessageBox, NULL, addr str_heading, addr str_hello, MB_OK--使用 invoke 伪指令调用 MessageBox，将地址为 str_heading 的内容作为对话框的标题，将地址为 str_hello 的内容作为对话框的内容，并将对话框显示在屏幕上。
    invoke ExitProcess, 0--调用结束进程的函数，退出进程
END start--结束
```

具体实验操作过程

- 先编写好两个程序的汇编代码，然后将后缀改为 .asm。
- 在当前文件目录下打开命令提示符 cmd。
- 使用 ml.exe 对汇编代码进行汇编但不链接，在本目录下得到相应的 .obj 文件。

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab1>.\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
************
```

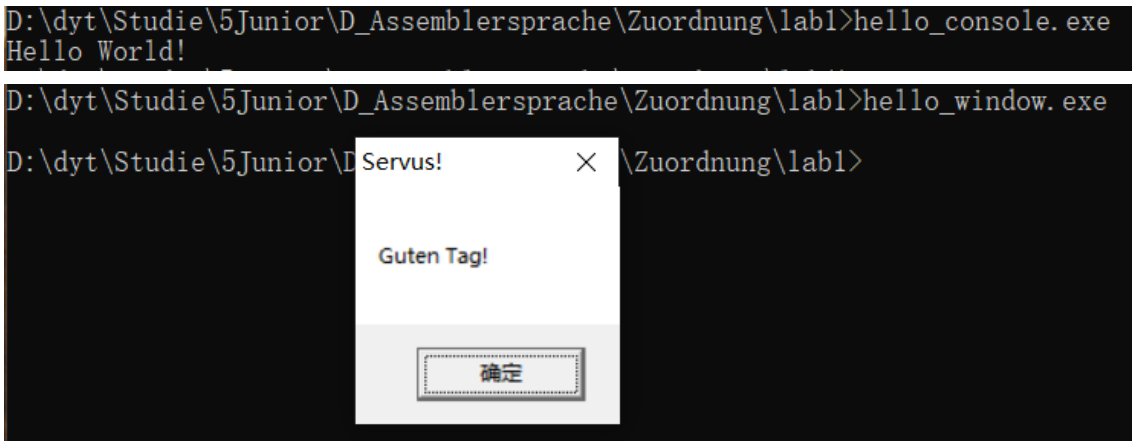
```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab1>.\masm32\bin\ml /c /Zd /coff hello_window.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: hello_window.asm

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

- 使用命令对文件进行链接，得到 .exe 可执行文件。

```
D:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab1>.\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:\dyt\Studie\5Junior\D_Assemblersprache\Zuordnung\lab1>.\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.
```

- 运行可执行文件 hello_console.exe 和 hello_window.exe。



参考文献/网站

1.ASMA32 官方网站 http://www.asma32.com/

2.汇编程序之windows console例子解说 https://www.cnblogs.com/qnbs1/articles/1713506.html