



南開大學
Nankai University

计算机学院
汇编语言与逆向技术实验报告

Lab1-HelloWorld

姓名：杨冰雪

学号：2110508

专业：计算机科学与技术

2023 年 10 月 13 日

目录

1 实验内容	2
2 实验步骤	2
2.1 编辑	2
2.2 编译	3
2.3 连接	3
2.4 执行	3
3 实验截图	3
3.1 程序一	3
3.2 程序二	4
4 代码解析	5
4.1 命令和参数解析	5
4.2 汇编程序解析	6
4.2.1 汇编程序 1: hello_console.asm	6
4.2.2 汇编程序 2: hello_window.asm	7
5 实验总结	9

1 实验内容

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，首先需要使用汇编程序将汇编文件转换成目标文件，再使用连接程序将目标文件连接成可执行文件。

汇编程序 1: hello_console.asm

```
1 .386
2 .model flat, stdcall
3 option casemap :none
4 include \masm32\include\windows.inc
5 include \masm32\include\kernel32.inc
6 include \masm32\include\masm32.inc
7 includelib \masm32\lib\kernel32.lib
8 includelib \masm32\lib\masm32.lib
9 .data
10 str_hello BYTE "Hello World!", 0
11 .code
12 start:
13 invoke StdOut, addr str_hello
14 invoke ExitProcess, 0
15 END start
```

汇编程序 2: hello_window.asm

```
1 .386
2 .model flat, stdcall
3 option casemap :none
4 include \masm32\include\windows.inc
5 include \masm32\include\kernel32.inc
6 include \masm32\include\user32.inc
7 includelib \masm32\lib\kernel32.lib
8 includelib \masm32\lib\user32.lib
9 .data
10 str_hello BYTE "Hello World!", 0
11 .code
12 start:
13 invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK
14 invoke ExitProcess, 0
15 END start
```

2 实验步骤

2.1 编辑

使用编辑软件（Notepad）形成源程序（.asm），如：hello_console.asm 和 hello_window.asm。

2.2 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：

```
1  "\"masm32\bin\ml /c /Zd /coff hello_console.asm"
2  "\"masm32\bin\ml /c /Zd /coff hello_window.asm"
```

2.3 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：
代码样式 1

```
1  "\"masm32\bin\Link /SUBSYSTEM:CONSOLE hello_console.obj"
2  "\"masm32\bin\Link /SUBSYSTEM:WINDOWS hello_window.obj"
```

2.4 执行

直接执行生成的可执行文件，在屏幕上显示结果。

3 实验截图

3.1 程序一

1. 在 MASM32 Editor 程序中输入汇编代码，并保存文件为 hello_console.asm。

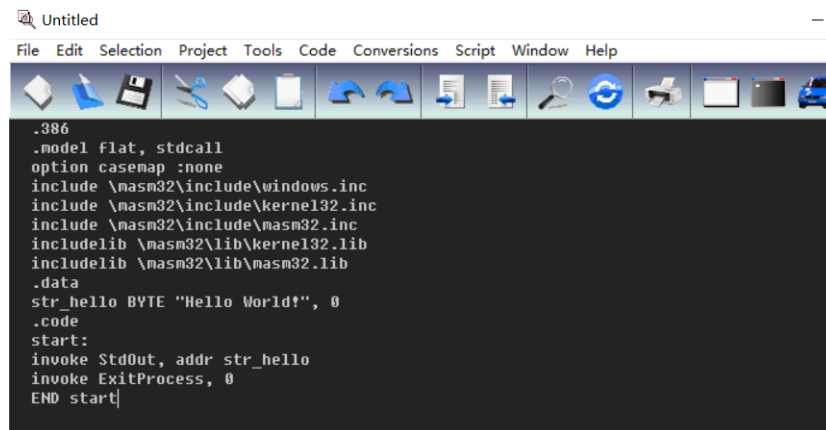


图 3.1: 汇编文件

2. 打开终端，执行命令将刚保存到汇编文件编译为目标文件。

```
E:\masm_code\lab1>D:\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
*****
```

图 3.2: 编译过程

3. 执行连接命令将目标文件连接成可执行文件。

```
E:\masm_code\lab1>D:\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.
```

图 3.3: 连接过程

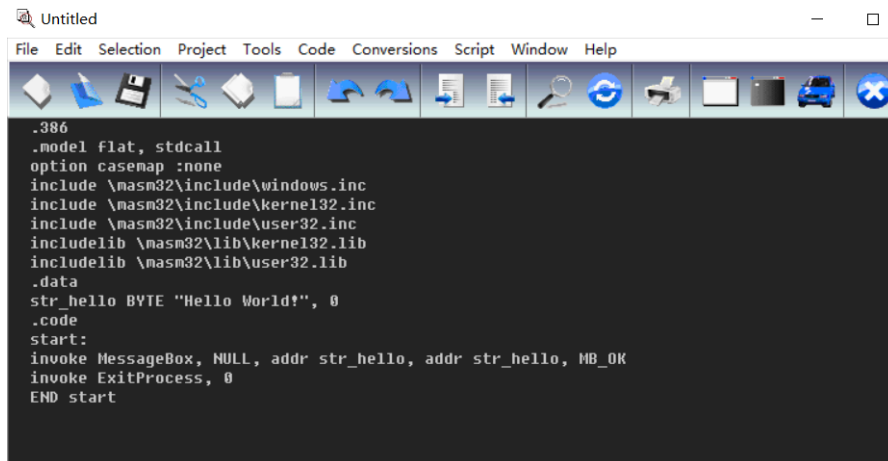
4. 运行可执行文件，在显示屏上观察到命令行中输出了” Hello World! ”这句话。

```
E:\masm_code\lab1>.hello_console
Hello World!
E:\masm_code\lab1>_
```

图 3.4: 执行可执行文件

3.2 程序二

1. 在 MASM32 Editor 程序中输入汇编代码，并保存文件为 hello_window.asm。



```
Untitled
File Edit Selection Project Tools Code Conversions Script Window Help
[Icons]
.asm32
.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
```

图 3.5: 汇编文件

2. 打开终端，执行命令将刚保存到汇编文件编译为目标文件。

```
E:\masm_code\lab1>D:\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
*****
```

图 3.6: 编译过程

3. 执行连接命令将目标文件连接成可执行文件。

```
E:\masm_code\lab1>D:\masm32\bin\Link /SUBSYSTEM:CONSOLE hello_window.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

图 3.7: 连接过程

4. 运行可执行文件，在显示屏上观察到在 Windows MessageBox 中输出了” Hello World! ”这句话。

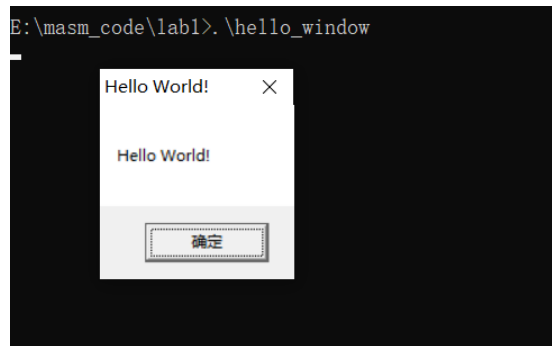


图 3.8: 执行可执行文件

4 代码解析

4.1 命令和参数解析

因为汇编程序一和汇编程序二的命令相同，只是文件的名字不同，所以下面就只以第一个汇编程序的命令为例，进行分析。

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

- \masm32\bin\ml 用来汇编和连接一个或多个汇编程序。
- /c 表示只编译，不连接。
- /Zd 表示在目标文件中生成行号信息
- /coff 表示生成 Microsoft 公共目标文件格式（common object file format）的文件（.obj 文件）
- hello_console.asm 表示要生成的目标文件

```
1  "\masm32\bin\Link /SUBSYSTEM:CONSOLE hello_console.obj"
```

- \masm32\bin\link 表示将 obj 文件进行连接，生成可执行文件。
- /SUBSYSTEM:CONSOLE 表示生成命令行程序
- hello_console.obj 表示目标文件

4.2 汇编程序解析

4.2.1 汇编程序 1: hello_console.asm

汇编程序 1: hello_console.asm

```
1  .386
2  .model flat, stdcall
3  option casemap :none
4  include \masm32\include\windows.inc
5  include \masm32\include\kernel32.inc
6  include \masm32\include\masm32.inc
7  includelib \masm32\lib\kernel32.lib
8  includelib \masm32\lib\masm32.lib
9  .data
10 str_hello BYTE "Hello World!", 0
11 .code
12 start:
13 invoke StdOut, addr str_hello
14 invoke ExitProcess, 0
15 END start
```

- .386
这条指令的作用用来编译器使用.386 指令集。
- .model flat, stdcall
.model 是指定程序的内存模式的汇编指令。flat 是一种方便的系统程序模式，在这种模式下不再区分远指针 (far) 和近指针 (near)。Stdcall 是一种系统函数传递参数的方法，表示从右到左的顺序传递参数。
- option casemap :none
强制程序代码大小写敏感。
- include \masm32\include\windows.inc
系统程序必需的包含文件。windows.inc 通常包含了 Win32 API 常量和定义的声明。
- include \masm32\include\kernel32.inc
kernel32.inc 包含了我们所使用的 ExitProcess 函数。
- include \masm32\include\masm32.inc
masm32.inc 包含了 StdOut 函数。

- `includelib \masm32\lib\kernel32.lib`
函数依赖库。
- `includelib \masm32\lib\masm32.lib`
函数依赖库。
- `.data`
定义程序已初始化数据段。
- `str_hello BYTE "Hello World!", 0`
声明变量 `str_hello`，其大小为字节 (BYTE)，其值为 "Hello World!"，后面跟着一个 "NULL" 字母，这是因为 ANSI 字符串必须以 NULL 结尾。
- `.code`
定义程序代码段。
- `start:`
指令标号，程序的代码位于这个标号的后面。
- `invoke StdOut, addr str_hello`
调用 `StdOut` 函数并传入参数为 `str_hello` 的地址，从而将内存数据输出到命令行窗口上。
- `invoke ExitProcess, 0`
调用 `ExitProcess` 函数并传入参数为 0，从而退出程序。
- `END start`
标记程序的结束。

4.2.2 汇编程序 2: `hello_window.asm`

汇编程序 2: `hello_window.asm`

```
1 .386
2 .model flat, stdcall
3 option casemap :none
4 include \masm32\include\windows.inc
5 include \masm32\include\kernel32.inc
6 include \masm32\include\user32.inc
7 includelib \masm32\lib\kernel32.lib
8 includelib \masm32\lib\user32.lib
9 .data
10 str_hello BYTE "Hello World!", 0
11 .code
12 start:
13 invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK
14 invoke ExitProcess, 0
15 END start
```


- `.386`
这条指令的作用用来编译器使用 `.386` 指令集。
- `.model flat, stdcall`
`.model` 是指定程序的内存模式的汇编指令。`flat` 是一种方便的系统程序模式，在这种模式下不再区分远指针 (`far`) 和近指针 (`far`)。`Stdcall` 是一种系统函数传递参数的方法，表示从右到左的顺序传递参数。
- `option casemap :none`
强制程序代码大小写敏感。
- `include \masm32\include\windows.inc`
系统程序必需的包含文件。`windows.inc` 通常包含了 Win32 API 常量和定义的声明。
- `include \masm32\include\kernel32.inc`
`kernel32.inc` 包含了我们所使用的 `ExitProcess` 函数。
- `include \masm32\include\user32.inc`
`user32.inc` 包含了 `MessageBox` 函数。
- `includelib \masm32\lib\kernel32.lib`
函数依赖库。
- `includelib \masm32\lib\user32.lib`
函数依赖库。
- `.data`
定义程序已初始化数据段。
- `str_hello BYTE "Hello World!", 0`
声明变量 `str_hello`，其大小为字节 (`BYTE`)，其值为 "Hello World!"，后面跟着一个 "NULL" 字母，这是因为 ANSI 字符串必须以 NULL 结尾。
- `.code`
定义程序代码段。
- `start:`
指令标号，程序的代码位于这个标号的后面。
- `invoke MessageBox, NULL, addr str_hello, addr str_hello, MB_OK`
调用 `MessageBox`，并传入四个参数，一个参数 `NULL` 表示该消息框没有所有者窗口，第二个参数 `addr str_hello` 表示要显示的信息为 "Hello World!"，第三个参数 `addr str_hello` 表示消息框的标题为 "Hello World!"，第四个参数 `MB_OK` 表示消息框包含一个确定按钮。
- `invoke ExitProcess, 0`
调用 `ExitProcess` 函数并传入参数为 0，从而退出程序。
- `END start`
标记程序的结束。

5 实验总结

通过本次实验，学会了如何安装 masm 程序，了解了如何通过使用 masm 的编译器来将汇编程序转化为可执行程序的过程，学会更加熟练使用终端输入命令，同时也对汇编语言有了更深刻的理解，熟悉了汇编文件的格式和框架。