

目 录

第一章 PE 学习环境搭建	3
1.1 源码编译工具	3
1.1.1 汇编语言开发环境	3
1.1.2 C 语言开发环境	错误!未定义书签。
1.2 静态分析工具	错误!未定义书签。
1.2.1 WinHex	错误!未定义书签。
1.2.2 IDA	错误!未定义书签。
1.3 动态调试工具	错误!未定义书签。
1.3.1 CE 内存搜索工具	错误!未定义书签。
1.3.2 DTDebug 调试器	错误!未定义书签。
1.3.3 x64dbg 调试器	错误!未定义书签。
第二章 PE 文件	错误!未定义书签。
2.1 初识 PE 文件	错误!未定义书签。
2.1.1 PE 文件特征和组成	错误!未定义书签。
2.1.2 初识 PE 文件	错误!未定义书签。
2.2 加载 PE 文件	错误!未定义书签。
2.2.1 加载 PE 文件的过程	错误!未定义书签。
2.2.2 加载 PE 文件	错误!未定义书签。
第三章 PE 头结构	错误!未定义书签。
3.1 基本概念	错误!未定义书签。
3.1.1 地址	错误!未定义书签。
3.1.2 指针	错误!未定义书签。
3.1.3 数据目录项	错误!未定义书签。
3.1.4 节	错误!未定义书签。
3.1.5 对齐方式	错误!未定义书签。
3.1.6 字符串编码格式	错误!未定义书签。
3.2 PE 头结构	错误!未定义书签。
3.2.1 DOS 头	错误!未定义书签。
3.2.2 DOS Stub	错误!未定义书签。
3.2.3 NT 头	错误!未定义书签。
3.2.4 节表	错误!未定义书签。
3.3 PE 内存映像	错误!未定义书签。
3.3.1 PE 文件读进内存的两种方法	错误!未定义书签。

3.3.2 加载到内存中的 PE	错误!未定义书签。
第四章 导入表	错误!未定义书签。
4.1 导入表	错误!未定义书签。
4.1.1 导入表数据结构	错误!未定义书签。
4.1.2 PE 中的导入表	错误!未定义书签。
4.1.3 IAT 函数地址表	错误!未定义书签。
4.1.4 手工重构导入表	错误!未定义书签。
4.2 绑定导入表	错误!未定义书签。
4.3 延迟加载导入表	错误!未定义书签。
第五章 导出表	错误!未定义书签。
第六章 PE 重定位表	错误!未定义书签。
第七章 资源表	错误!未定义书签。
第八章 其他常见节表	错误!未定义书签。
第九章 PE 变形	错误!未定义书签。
第十章 PE 工具	错误!未定义书签。
3.3.2 PElLoader	错误!未定义书签。
第十一章 动态加载	错误!未定义书签。
第十二章 拷贝节表	错误!未定义书签。
第十三章 PE 拷贝节	错误!未定义书签。
第十四章 PE 添加节	错误!未定义书签。
第十五章 PE 补丁	错误!未定义书签。
第十六章 PE 捆绑	错误!未定义书签。
第十七章 PE 病毒	错误!未定义书签。
第十八章 加密	错误!未定义书签。
第十九章 破解	错误!未定义书签。

第一章 PE 学习环境搭建

在正式学习 PE 之前，先给大家介绍一下学习 WindowsPE 所需的开发环境和相关工具软件的使用方法。

本章学习知识概要：

- 源码编译工具
- 静态分析工具
- 动态分析工具

1.1 源码编译工具

本书提供的示例代码有两种，一种是 32 位汇编语言代码，另一种是 32 位 C 语言代码。因此，需要使用汇编语言和 C 语言两种不同的编译工具。接下来我们将介绍汇编和 C 语言两种编译环境的搭建。有兴趣的读者可以将两种代码互译。

本节必须掌握的知识点：

- ◆ 汇编语言开发环境
- ◆ C 语言开发环境

1.1.1 汇编语言开发环境

■ MASM32 介绍

MASM32 是 Steve Hutchesson 在微软的不同产品基础上集成开发出来的汇编开发工具包，适合 Win32 编程环境的汇编语言，主要用于基于 Windows 平台的 32 位汇编语言开发，是现在最流行的 Win32 汇编开发包。与 VC++ 和 VB 等高级语言相比，Win32 汇编具有得天独厚的优势，这些优势主要体现在：

(1) 摒弃了对系统细节的封装，更接近于系统的底层，从而使得编码更加灵活，能完成许多高级语言无法做到的事情（如代码重定位和特殊寄存器赋值等）。

(2) 生成的可执行 PE 文件体积小，执行速度快。

(3) 可用于软件的核心程序段设计，以提高软件的性能。

(4) 能够直接接触系统的底层，所以使用它要远比使用 VC++ 和 VB 等高级语言更适合开发与系统安全相关的程序。比如，与计算机硬件密切相关的驱动程序的开发、计算机病毒的分析与防治、软件加密与解密、软件调试、Windows PE 研究等。

MASM32 是一个免费的软件包，该软件包中包含了汇编器 ml.exe、资源编译器 rc.exe、32 位的链接器 link.exe 和一个简单的集成开发环境（Integrated Development Environment, IDE）QEditor.exe。

软件包中的 ml.exe 来自 Microsoft 的 MASM 软件包，rc.exe 和 link.exe 则来自 Microsoft 的 Visual Studio。

MASM32 软件包还包括了详尽的头文件、导入库文件、例子文件、帮助文档和一些工具程序，如 lib.exe 和 dumpbin.exe 等。可以从网站 <http://www.masm32.com/> 上获得 MASM32 SDK 的最新版本（masm32v11r），并可以在论坛里与来自世界各地的汇编爱好者交流技术和思想。

■ MASM32 安装

● **步骤 1:** 运行安装程序 install.exe，安装汇编环境。

首先选择安装路径，此处我们选择的路径为 D:\(软件)，然后单击“Install”按钮，如图 1-1 和图 1-2 所示。此后，中间过程所有的按钮均选择默认的设置，即可完成软件安装。安装结束后，会显示 IDE 汇编集成环境 QEditor 的界面，如图 1-3 所示。我们也可以选择自己熟悉的文本编辑工具，建议使用 Notepad++，如图 1-4 所示。



图 1-1 MASM32 安装

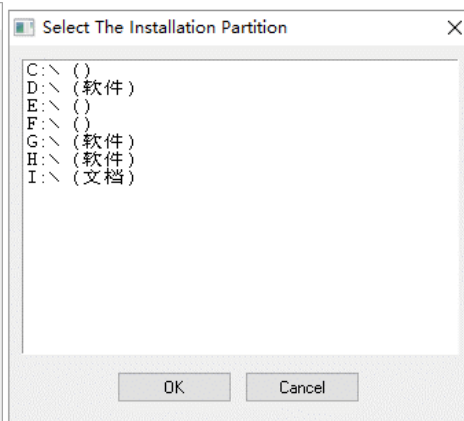


图 1-2 安装路径

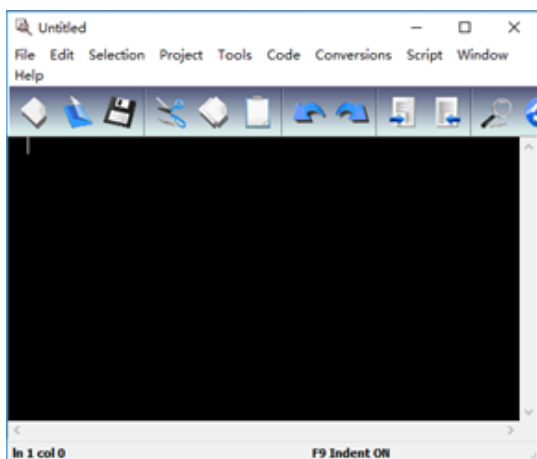


图 1-3 Qeditor

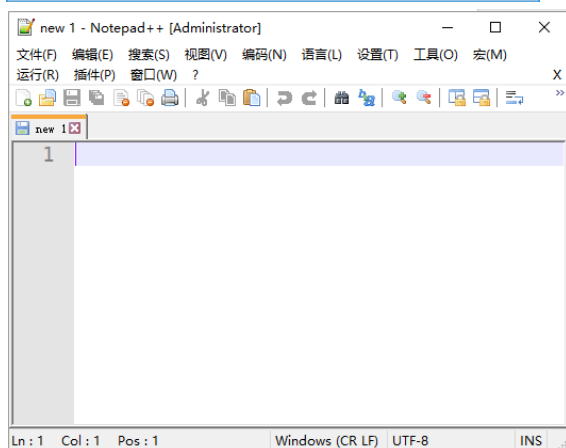


图 1-4 Notepad++

●步骤 2: 建立自己的工作区

建议选择将软件安装到非系统盘（例如 D 盘）。为了能存放自己编写的汇编代码。可以在 D:\masm32 中新建立一个文件夹 source 存放代码。也可以在其他目录新建一个文件夹，如图 1-5 所示。接下来设置汇编程序所依赖的环境，以及要调用的 API 函数库都在该文件夹内。

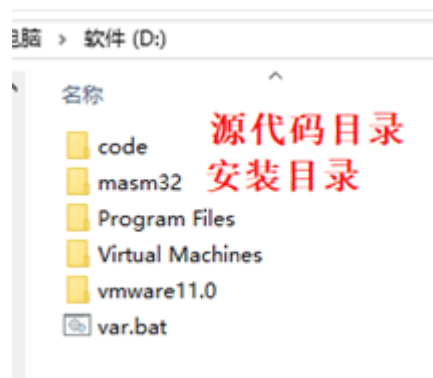


图 1-5 设置工作区

●步骤 3 设置系统环境变量

“我的电脑”上点鼠标右键，选择“属性”，选择“高级”选项卡，单击“环境变量”按钮，在用户的环境变量中增加以下三个环境变量，如图 1-6 所示：

1. include=d:\masm32\include //头文件目录
2. lib=d:\masm32\lib //lib 库
3. path=d:\masm32\bin //编译工具软件

如果系统中已经存在相同名字的环境变量（如 path 变量），则在该变量的值的最后加上一个分号，然后加上上面列出的值即可。

例如，假设未设置前系统存在路径变量 path，且值为：
C:\Users\16400\AppData\Local\Microsoft\WindowsApps

修改以后的值为：

d:\masm32\bin; C:\Users\16400\AppData\Local
\Microsoft\WindowsApps

特别提示：要用英文输入，且不要忘记前面的分号。

也可以建立系统环境变量的批处理文件：假设文件名为 var.bat

```
@echo off
set include = d:\masm32\include
set lib = d:\masm32\lib
set path = d:\masm32\bin;%path%
echo on
```

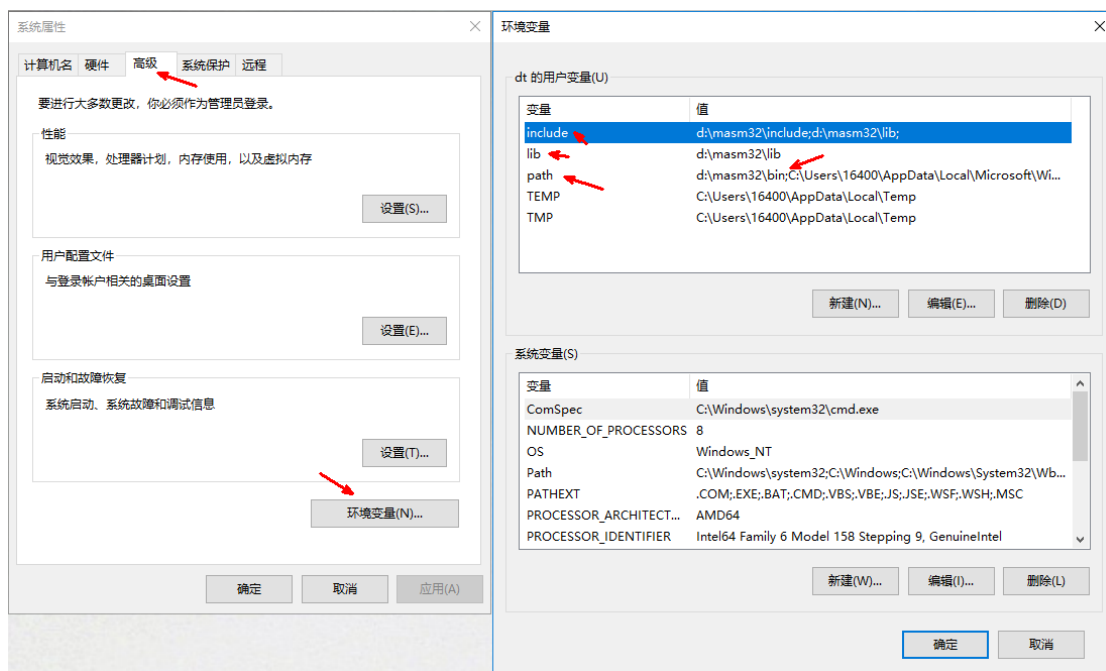


图 1-6 环境变量设置

文件中设置了 3 个环境变量：

1. `include` 变量指定头文件的搜索目录。定义了这个环境变量后，`ml.exe` 和 `rc.exe` 在处理 `asm` 和 `rc` 文件中遇到 `include` 语句时，会自动在环境变量定义的目录中查找 `include` 语句指定的文件，这样 `include` 语句中就不必写头文件的全路径名，如下所示：

```
include c:\masm32\include\windows.inc //不设置 include 环境变量时的写法
include windows.inc //设置 include 环境变量后可以这样写
```

这样处理的好处是以后移动了 MASM32 的安装位置后不必修改每个源文件中的 `include` 语句。

如果使用 Visual C++ 的集成环境来建立 `rc` 文件的话，为了使 `rc.exe` 能找到头件，还要把 VC++ 安装目录下的 `include` 和 `MFC\include` 目录包含进来，多个路径之间用“;”隔开：

```
set include=x:\masm32\include;VC 目录\include;VC 目录\MFC\include。
```

2. `lib` 环境变量指定导入库文件的搜索目录。`ml.exe` 根据这个变量寻找 `include lib` 语句指定的导入库文件，`Link.exe` 也根据这个变量寻找库文件的位置。

3. `path` 环境变量可以使我们不必在键入命令时带长长的路径。

■**代码编辑工具软件 2**：使用 `Notepad++`，如图 1-7 所示。可以根据个人偏好选择自己满意的编辑工具。

实验一：使用 `NotePad++` 编写 32 位汇编源程序 `HelloWord.asm`。

```

D:\code\masm32\2.HelloWorld.asm - Notepad++ [Administrator]
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
test.asm [2] 2.HelloWorld.asm [2]
1 ;FileName:hello.asm
2 ;例1: hello
3 ;by:bcdaren
4 ;2021.01.06
5 ;=====
6
7 .386
8 .model flat,stdcall
9 option casemap:none ;区分大小写
10
11 include windows.inc
12 include user32.inc
13 includelib user32.lib
14 include kernel32.inc
15 includelib kernel32.lib
16
17 .data
18 szCaption byte "hello",0
19 szText byte "Hello,welcome to win32 asm!",0
20
21 .code
22 start:
23     invoke MessageBox,NULL,offset szText,offset szCaption,MB_OK
24     invoke ExitProcess,NULL
25 end start
26
Assembly language length: 482 lines: 26 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

```

图 1-7 Notepad++

■ 源代码

```

;FileName:HelloWorld.asm
;例 1: 第一个 32 位汇编源程序
;by:bcdaren
;2021.01.06
;=====
.386 ;支持 386 及以上 CPU
.model flat,stdcall ;flat 内存模式, stdcall 调用约定
option casemap:none ;区分大小写
;Windows 系统头文件和导入库
include windows.inc
include user32.inc
includelib user32.lib
include kernel32.inc
includelib kernel32.lib
;数据段简化定义
.data
szCaption byte "hello",0
szText byte "Hello,welcome to PE!",0
;代码段简化定义
.code
start:
;显示窗口信息
invoke MessageBox,NULL,offset szText,offset szCaption,MB_OK
invoke ExitProcess,NULL;退出程序, 返回操作系统
end start;程序入口地址

```

参见本书源代码 ch01。

■ 命令行编译方式

● 步骤 1: 进入工作区

通过转换磁盘命令和 CD 命令进入存放源文件 HelloWorld.asm 的目录中, 命令如下:

```
C:\Users\16400>d:
```

```
D:\>cd D:\code\winpe\ch01
```

```
D:\code\winpe\ch01>
```

● 步骤 2: 编译源文件

在当前工作区中输入命令:

“ml -c -coff HelloWorld.asm” 或者 “ml /c /coff HelloWorld.asm”, 然后回车。

```
D:\code\masm32\Chapter28\HelloWorld>ml -c -coff HelloWorld.asm
```

```
Microsoft (R) Macro Assembler Version 6.14.8444
```

```
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
```

```
Assembling: HelloWorld.asm
```

```
*****
```

```
ASCII build
```

```
*****
```