



南開大學
Nankai University

计算机学院
汇编语言与逆向技术实验报告

实验 5 peviewer

姓名：杨冰雪
学号：2110508
专业：计算机科学与技术

2023 年 11 月 22 日

目录

1 实验目的	2
2 实验环境	2
3 实验原理	2
3.1 PE 文件结构	2
3.2 Windows 文件读操作	4
4 实验过程	5
4.1 peviewer 程序的设计说明	5
4.2 peviewer 程序控制流图	6
4.3 peviewer.asm 源代码	6
4.4 peviewer.exe 运行结果	14
4.4.1 编译	14
4.4.2 链接	14
4.4.3 执行	15

1 实验目的

1. 熟悉 PE 文件结构；
2. 使用 Windows API 函数读取文件内容

2 实验环境

Windows 操作系统，MASM32 编译环境

3 实验原理

3.1 PE 文件结构

PE 文件结构如图所示。二进制 PE 文件包括：DOS 部首、PE 文件头、块表 (Section Table)、块 (Section)、调试信息 5 个部分。

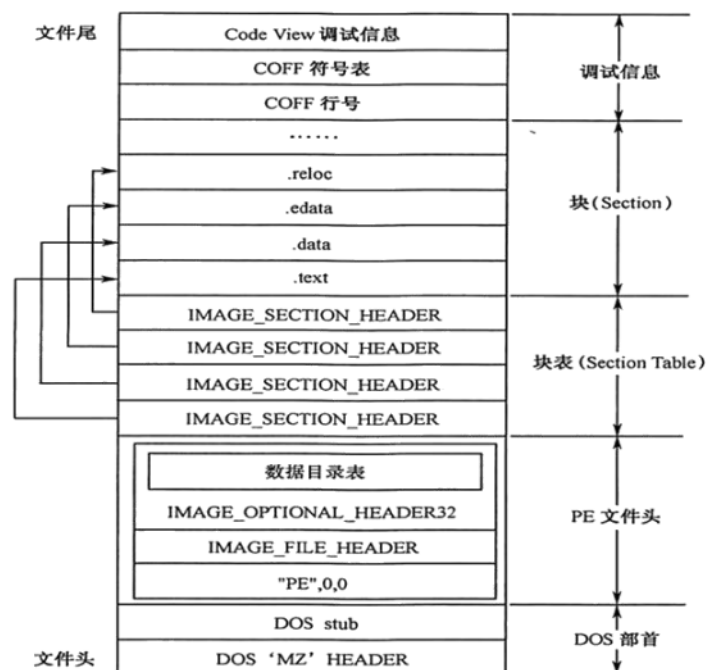


图 1 PE 文件结构

图 3.1: PE 文件结构

- DOS 部首是 DOS 系统的残留内容，目的是防止 Windows 系统的可执行程序在 DOS 系统上执行时导致 DOS 系统崩溃。DOS 部首是一段 DOS 程序，输出一段提示信息，说明程序只能运行在 Windows 系统上，不能运行在 DOS 系统上。

```

IMAGE_DOS_HEADER_STRUCT{
+0h  e_magic      WORD ?      ;DOS 可执行文件标记 "MZ"
+2h  e_cblp       WORD ?
+4h  e_cp         WORD ?
+6h  e_crlc       WORD ?
+8h  e_cparhdr    WORD ?
+0ah  e_minalloc   WORD ?
+0ch  e_maxalloc   WORD ?
+0eh  e_ss        WORD ?
+10h  e_sp        WORD ?
+12h  e_csum      WORD ?
+14h  e_ip        WORD ?      ;DOS 代码入口 IP
+16h  e_cs        WORD ?      ;DOS 代码入口 CS
+18h  e_lfarlc    WORD ?
+1ah  e_ovno      WORD ?
+1ch  e_res       WORD 4 dup(?)
+24h  e_oemid     WORD ?
+26h  e_oeminfo   WORD ?
+28h  e_res2      WORD 10 dup(?)
+3ch  e_lfanew    DWORD ?      ;指向 PE 文件头 "PE", 0, 0
} IMAGE_DOS_HEADER_ENDS

```

图 3.2: DOS 头数据结构

- PE 文件头记录了各种文件的装载信息，有映像的基地址 (ImageBase)、程序的入口地址 (Entry-Point)、数据块、编译时间、运行平台、数据目录表等信息。PE 文件头包括 Signature、FileHeader、OptionalHeader 三部分，数据结构如下所示：

```

1  IMAGE_NT_HEADERS STRUCT
2      +0h Signature DWORD
3      +4h FileHeader IMAGE_FILE_HEADER
4      +18h OptionalHeader IMAGE_OPTIONAL_HEADER32
5  IMAGE_NT_HEADERS ENDS

```

– Signature 的定义是 IMAGE_NT_HEADER，值为 00004550h

– FileHeader 的数据结构如下所示：

```

1  IMAGE_FILE_HEADER STRUCT
2      +04h Machine WORD ?
3      +06h NumberOfSections WORD ?
4      +08h TimeDateStamp DWORD ?
5      +0Ch PointerToSymbolTable DWORD ?
6      +10h NumberOfSymbols DWORD ?
7      +14h SizeOfOptionalHeader WORD ?
8      +16h Characteristics WORD ?

```

– OptionalHeader 的数据结构如下所示：

```

1  IMAGE_OPTIONAL_HEADER STRUCT
2      +18h Magic WORD ?
3      +1Ah MajorLinkerVersion BYTE ?
4      +1Bh MinorLinkerVersion BYTE ?
5      +1Ch SizeOfCode DWORD ?

```

```

6      +20h SizeOfInitializedData DWORD ?
7      +24h SizeOfUninitializedData DWORD ?
8      +28h AddressOfEntryPoint DWORD ?
9      +2Ch BaseOfCode DWORD ?
10     +30h BaseOfData DWORD ?
11     +34h ImageBase DWORD ?
12     +38h SectionAlignment DWORD ?
13     +3Ch FileAlignment DWORD ?
14     +40h MajorOperatingSystemVersion WORD ?
15     +42h MinorOperatingSystemVersion WORD ?
16     +44h MajorImageVersion WORD ?
17     +46h MinorImageVersion WORD ?
18     +48h MajorSubsystemVersion WORD ?
19     +4Ah MinorSubsystemVersion WORD ?
20     +4Ch Win32VersionValue DWORD ?
21     +50h SizeOfImage DWORD ?
22     +54h SizeOfHeaders DWORD ?
23     +58h CheckSum DWORD ?
24     +5Ch Subsystem WORD ?
25     +5Eh DllCharacteristics WORD ?
26     +60h SizeOfStackReserve DWORD ?
27     +64h SizeOfStackCommit DWORD ?
28     +68h SizeOfHeapReserve DWORD ?
29     +6Ch SizeOfHeapCommit DWORD ?
30     +70h LoaderFlags DWORD ?
31     +74h NumberOfRvaAndSizes DWORD ?
32     +78h DataDirectory

```

- 块表 (Section Table) 描述代码块、数据块、资源块等不同数据块的文件和内存的映射，数据块的各种属性。
- 块 (Section) 分别存储了程序的代码、数据、资源等信息。

3.2 Windows 文件读操作

读一个文件用到的 Windows API 函数有 CreateFile、SetFilePointer、ReadFile、CloseHandle。

- CreateFile 函数的原型如下：

```

1  HANDLE CreateFile(
2      LPCTSTR lpFileName,
3      DWORD dwDesiredAccess,
4      DWORD dwShareMode,
5      LPSECURITY_ATTRIBUTES lpSecurityAttributes,
6      DWORD dwCreationDisposition,

```

```
7     DWORD dwFlagsAndAttributes,  
8     HANDLE hTemplateFile  
9 );
```

- SetFilePointer 函数原型如下:

```
1  DWORD SetFilePointer(  
2      HANDLE hFile,  
3      LONG lDistanceToMove,  
4      PLONG lpDistanceToMoveHigh,  
5      DWORD dwMoveMethod  
6  );
```

- ReadFile 函数原型如下:

```
1  BOOL ReadFile(  
2      HANDLE hFile,  
3      LPVOID lpBuffer,  
4      DWORD nNumberOfBytesToRead,  
5      LPDWORD lpNumberOfBytesRead,  
6      LPOVERLAPPED lpOverlapped  
7  );
```

- CloseHandle 函数原型如下:

```
1  BOOL CloseHandle(  
2      HANDLE hObject  
3  );
```

4 实验过程

4.1 peviewer 程序的设计说明

1. 定义数据段
2. 输入 PE 文件的文件名
3. 调用 Windows API 函数打开 PE 文件
 - 调用 CreateFile 函数, 利用用户输入的文件名打开相应的文件
 - 调用 SetFilePointer 函数, 设置文件指针
 - 调用 ReadFile 函数, 从指定位置读取文件内容
 - 调用 CloseHandle 函数, 关闭句柄
4. 将读取出来的内容通过 dw2hex 函数转换成 16 进制输出到命令行

5. 循环以上过程，依次读取 IMAGE_DOS_HEADER 结构中的 e_magic 和 e_lfanew 字段的值，PE 文件的 IMAGE_NT_HEADER 结构中的 Signature 字段的值，IMAGE_NT_HEADER 结构中的 IMAGE_FILE_HEADER 结构，从中读取出字段 NumberOfSections、TimeDateStamp、Characteristics 的值，读取 IMAGE_NT_HEADER 结构中的 IMAGE_OPTIONAL_HEADER 结构，从中读取字段 AddressOfEntryPoint、ImageBase、SectionAlignment、FileAlignment 的值。
6. 调用 ExitProcess 函数，关闭程序。

4.2 peviewer 程序控制流图

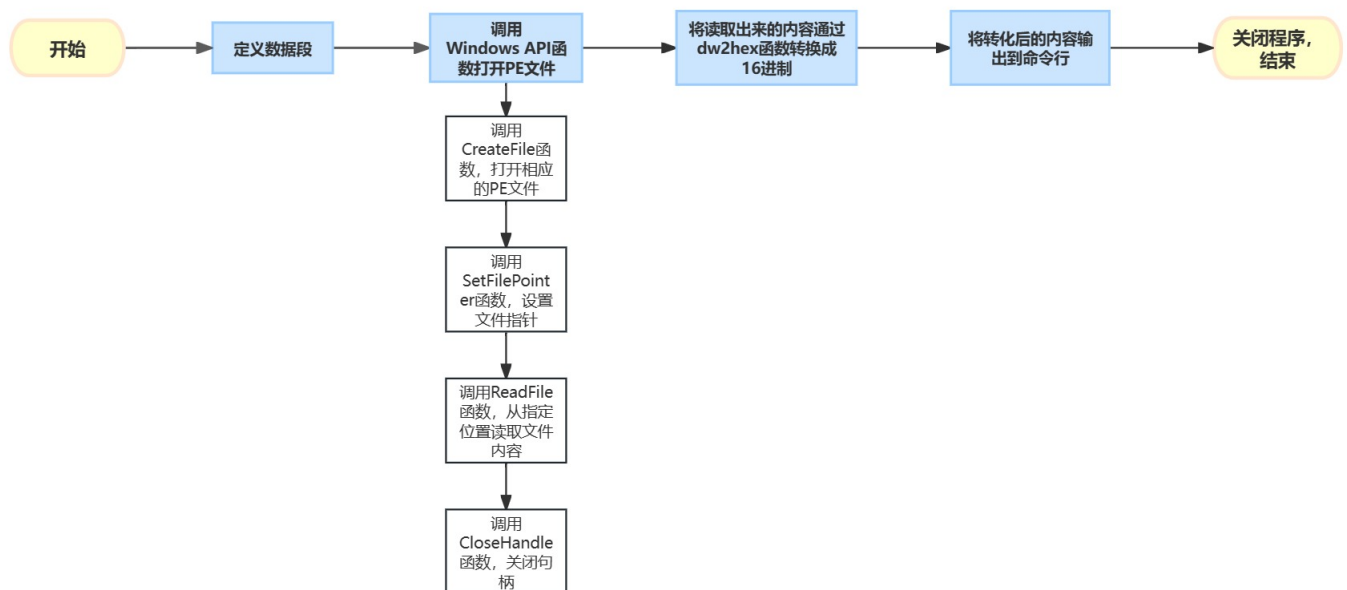


图 4.3: 程序控制流图

4.3 peviewer.asm 源代码

peviewer.asm

```

1  .386
2  .model flat, stdcall
3  option casemap :none
4  include D:\masm32\include\windows.inc
5  include D:\masm32\include\kernel32.inc
6  include D:\masm32\include\masm32.inc
7  includelib D:\masm32\lib\masm32.lib
8  includelib D:\masm32\lib\kernel32.lib
9
10 .data
11     buffin db 20 DUP(0) ;从文件中获取的内容
12     buffout db 20 DUP(0) ;转换后输出的内容
13     fileName db 20 DUP(0) ;文件名
  
```

```
14     hFile HANDLE 0 ;文件句柄
15     point db 0 ;记录偏移
16
17 ;定义输出字符串
18     str0 db "Please input a PE file :",0
19     str1 db 0Ah,"IMAGE_DOS_HEADER",0
20     str11 db 0Ah,"    e_magic: ",0
21     str12 db 0Ah,"    e_lfanew: ",0
22     str2 db 0Ah,"IMAGE_NT_HEADERS",0
23     str21 db 0Ah,"    Signature: ",0
24     str3 db 0Ah,"IMAGE_FILE_HEADER",0
25     str31 db 0Ah,"    NumberOfSections: ",0
26     str32 db 0Ah,"    TimeDateStamp: ",0
27     str33 db 0Ah,"    Characteristics: ",0
28     str4 db 0Ah,"IMAGE_OPTIONAL_HEADER",0
29     str41 db 0Ah,"    AddressOfEntryPoint: ",0
30     str42 db 0Ah,"    ImageBase: ",0
31     str43 db 0Ah,"    SectionAlignment: ",0
32     str44 db 0Ah,"    FileAlignment: ",0
33
34 .code
35 start:
36     invoke StdOut, ADDR str0
37     invoke StdIn, ADDR fileName, 20
38
39     ;e_magic
40     ;调用函数CreateFile来打开PE文件
41     invoke CreateFile, ADDR fileName,\
42             GENERIC_READ,\
43             FILE_SHARE_READ,\
44             0,\
45             OPEN_EXISTING,\
46             FILE_ATTRIBUTE_ARCHIVE,\
47             0
48
49     ;设置文件句柄
50     mov hFile, eax
51     ;设置文件指针
52     invoke SetFilePointer, hFile,\
53             0,\
54             0,\
55             FILE_BEGIN
56     ;读取文件内容,从hfile指向的位置读取20个字节到buffin
57     invoke ReadFile, hFile,\
58             ADDR buffin,\
59             20,\
60             0,\
61             0
62
63     ;将buffin转化为16进制存储到buffout
```



```
63     mov eax, DWORD PTR buffin
64     invoke dw2hex, eax, ADDR buffout
65     ; 输出
66     invoke StdOut, ADDR str1
67     invoke StdOut, ADDR str11
68     invoke StdOut, ADDR [buffout+4]
69     ; 调用函数CloseHandle关闭句柄
70     invoke CloseHandle, hFile
71
72     ; e_lfanew
73     ; 调用函数CreateFile来打开PE文件
74     invoke CreateFile, ADDR fileName, \
75             GENERIC_READ, \
76             FILE_SHARE_READ, \
77             0, \
78             OPEN_EXISTING, \
79             FILE_ATTRIBUTE_ARCHIVE, \
80             0
81     ; 设置文件句柄
82     mov hFile, eax
83     ; 设置文件指针
84     invoke SetFilePointer, hFile, \
85             3Ch, \
86             0, \
87             FILE_BEGIN
88     ; 读取文件内容, 从hfile指向的位置读取20个字节到buffin
89     invoke ReadFile, hFile, \
90             ADDR buffin, \
91             20, \
92             0, \
93             0
94     ; 将buffin转化为16进制存储到buffout
95     mov eax, DWORD PTR buffin
96     mov DWORD PTR point, eax
97     invoke dw2hex, eax, ADDR buffout
98     ; 输出
99     invoke StdOut, ADDR str12
100    invoke StdOut, ADDR buffout
101    ; 调用函数CloseHandle关闭句柄
102    invoke CloseHandle, hFile
103
104    ; Signature
105    ; 调用函数CreateFile来打开PE文件
106    invoke CreateFile, ADDR fileName, \
107            GENERIC_READ, \
108            FILE_SHARE_READ, \
109            0, \
110            OPEN_EXISTING, \
111            FILE_ATTRIBUTE_ARCHIVE, \
```

```
112         0
113     ;设置文件句柄
114     mov hFile, eax
115     ;设置文件指针
116     invoke SetFilePointer, hFile,\
117         point,\
118         0,\
119         FILE_BEGIN
120     ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
121     invoke ReadFile, hFile,\
122         ADDR buffin,\
123         20,\
124         0,\
125         0
126     ;将buffin转化为16进制存储到buffout
127     mov eax, DWORD PTR buffin
128     invoke dw2hex, eax, ADDR buffout
129     ;输出
130     invoke StdOut, ADDR str2
131     invoke StdOut, ADDR str21
132     invoke StdOut, ADDR buffout
133     ;调用函数CloseHandle关闭句柄
134     invoke CloseHandle, hFile
135
136     ;NumberOfSections
137     add point, 06h ;改变point的偏移值
138     ;调用函数CreateFile来打开PE文件
139     invoke CreateFile, ADDR fileName,\
140         GENERIC_READ,\
141         FILE_SHARE_READ,\
142         0,\
143         OPEN_EXISTING,\
144         FILE_ATTRIBUTE_ARCHIVE,\
145         0
146     ;设置文件句柄
147     mov hFile, eax
148     ;设置文件指针
149     invoke SetFilePointer, hFile,\
150         point,\
151         0,\
152         FILE_BEGIN
153     ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
154     invoke ReadFile, hFile,\
155         ADDR buffin,\
156         20,\
157         0,\
158         0
159     ;将buffin转化为16进制存储到buffout
160     mov eax, DWORD PTR buffin
```

```
161     invoke dw2hex, eax, ADDR buffout
162     ;输出
163     invoke StdOut, ADDR str3
164     invoke StdOut, ADDR str31
165     invoke StdOut, ADDR [buffout+4]
166     ;调用函数CloseHandle关闭句柄
167     invoke CloseHandle, hFile
168
169     ;TimeStamp
170     add point,02h ;改变point的偏移值
171     ;调用函数CreateFile来打开PE文件
172     invoke CreateFile, ADDR fileName,\
173             GENERIC_READ,\
174             FILE_SHARE_READ,\
175             0,\
176             OPEN_EXISTING,\
177             FILE_ATTRIBUTE_ARCHIVE,\
178             0
179     ;设置文件句柄
180     mov hFile, eax
181     ;设置文件指针
182     invoke SetFilePointer, hFile,\
183             point,\
184             0,\
185             FILE_BEGIN
186     ;读取文件内容,从hfile指向的位置读取20个字节到buffin
187     invoke ReadFile, hFile,\
188             ADDR buffin,\
189             20,\
190             0,\
191             0
192     ;将buffin转化为16进制存储到buffout
193     mov eax, DWORD PTR buffin
194     invoke dw2hex, eax, ADDR buffout
195     ;输出
196     invoke StdOut, ADDR str32
197     invoke StdOut, ADDR buffout
198     ;调用函数CloseHandle关闭句柄
199     invoke CloseHandle, hFile
200
201     ;Charateristics
202     add point,0Eh ;改变point的偏移值
203     ;调用函数CreateFile来打开PE文件
204     invoke CreateFile, ADDR fileName,\
205             GENERIC_READ,\
206             FILE_SHARE_READ,\
207             0,\
208             OPEN_EXISTING,\
209             FILE_ATTRIBUTE_ARCHIVE,\
```

```
210         0
211     ;设置文件句柄
212     mov hFile, eax
213     ;设置文件指针
214     invoke SetFilePointer, hFile,\
215             point,\
216             0,\
217             FILE_BEGIN
218     ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
219     invoke ReadFile, hFile,\
220             ADDR buffin,\
221             20,\
222             0,\
223             0
224     ;将buffin转化为16进制存储到buffout
225     mov eax, DWORD PTR buffin
226     invoke dw2hex, eax, ADDR buffout
227     ;输出
228     invoke StdOut, ADDR str33
229     invoke StdOut, ADDR buffout+4
230     ;调用函数CloseHandle关闭句柄
231     invoke CloseHandle, hFile
232
233     ;AddressOfEntryPoint
234     add point,12h ;改变point的偏移值
235     ;调用函数CreateFile来打开PE文件
236     invoke CreateFile, ADDR fileName,\
237             GENERIC_READ,\
238             FILE_SHARE_READ,\
239             0,\
240             OPEN_EXISTING,\
241             FILE_ATTRIBUTE_ARCHIVE,\
242             0
243     ;设置文件句柄
244     mov hFile, eax
245     ;设置文件指针
246     invoke SetFilePointer, hFile,\
247             point,\
248             0,\
249             FILE_BEGIN
250     ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
251     invoke ReadFile, hFile,\
252             ADDR buffin,\
253             20,\
254             0,\
255             0
256     ;将buffin转化为16进制存储到buffout
257     mov eax, DWORD PTR buffin
258     invoke dw2hex, eax, ADDR buffout
```

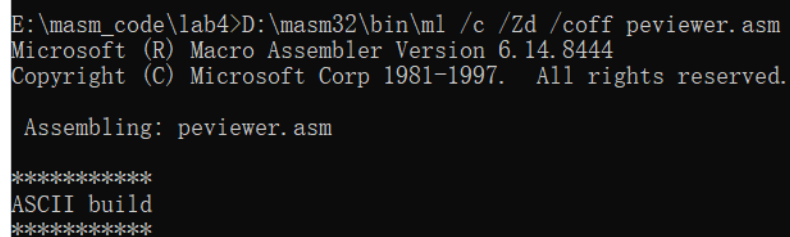
```
259 ;输出
260 invoke StdOut, ADDR str4
261 invoke StdOut, ADDR str41
262 invoke StdOut, ADDR buffout
263 ;调用函数CloseHandle关闭句柄
264 invoke CloseHandle, hFile
265
266 ;ImageBase
267 add point,0Ch ;改变point的偏移值
268 ;调用函数CreateFile来打开PE文件
269 invoke CreateFile, ADDR fileName,\
270             GENERIC_READ,\
271             FILE_SHARE_READ,\
272             0,\
273             OPEN_EXISTING,\
274             FILE_ATTRIBUTE_ARCHIVE,\
275             0
276 ;设置文件句柄
277 mov hFile, eax
278 ;设置文件指针
279 invoke SetFilePointer, hFile,\
280             point,\
281             0,\
282             FILE_BEGIN
283 ;读取文件内容,从hfile指向的位置读取20个字节到buffin
284 invoke ReadFile, hFile,\
285             ADDR buffin,\
286             20,\
287             0,\
288             0
289 ;将buffin转化为16进制存储到buffout
290 mov eax, DWORD PTR buffin
291 invoke dw2hex, eax, ADDR buffout
292 ;输出
293 invoke StdOut, ADDR str42
294 invoke StdOut, ADDR buffout
295 ;调用函数CloseHandle关闭句柄
296 invoke CloseHandle, hFile
297
298 ;SectionAlignment
299 add point,04h ;改变point的偏移值
300 ;调用函数CreateFile来打开PE文件
301 invoke CreateFile, ADDR fileName,\
302             GENERIC_READ,\
303             FILE_SHARE_READ,\
304             0,\
305             OPEN_EXISTING,\
306             FILE_ATTRIBUTE_ARCHIVE,\
307             0
```

```
308 ;设置文件句柄
309 mov hFile, eax
310 ;设置文件指针
311 invoke SetFilePointer, hFile,\
312             point,\
313             0,\
314             FILE_BEGIN
315 ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
316 invoke ReadFile, hFile,\
317             ADDR buffin,\
318             20,\
319             0,\
320             0
321 ;将buffin转化为16进制存储到buffout
322 mov eax, DWORD PTR buffin
323 invoke dw2hex, eax, ADDR buffout
324 ;输出
325 invoke StdOut, ADDR str43
326 invoke StdOut, ADDR buffout
327 ;调用函数CloseHandle关闭句柄
328 invoke CloseHandle, hFile
329
330 ;FileAlignment
331 add point,04h ;改变point的偏移值
332 ;调用函数CreateFile来打开PE文件
333 invoke CreateFile, ADDR fileName,\
334             GENERIC_READ,\
335             FILE_SHARE_READ,\
336             0,\
337             OPEN_EXISTING,\
338             FILE_ATTRIBUTE_ARCHIVE,\
339             0
340 ;设置文件句柄
341 mov hFile, eax
342 ;设置文件指针
343 invoke SetFilePointer, hFile,\
344             point,\
345             0,\
346             FILE_BEGIN
347 ;读取文件内容, 从hfile指向的位置读取20个字节到buffin
348 invoke ReadFile, hFile,\
349             ADDR buffin,\
350             20,\
351             0,\
352             0
353 ;将buffin转化为16进制存储到buffout
354 mov eax, DWORD PTR buffin
355 invoke dw2hex, eax, ADDR buffout
356 ;输出
```

```
357     invoke StdOut, ADDR str44
358     invoke StdOut, ADDR buffout
359     ;调用函数CloseHandle关闭句柄
360     invoke CloseHandle, hFile
361
362     invoke ExitProcess, 0
363 END start
```

4.4 peviewer.exe 运行结果

4.4.1 编译



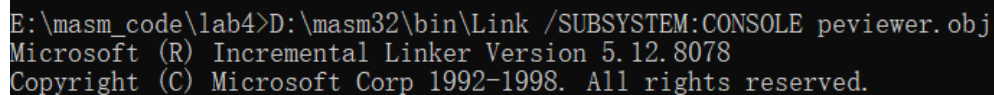
```
E:\masm_code\lab4>D:\masm32\bin\ml /c /Zd /coff peviewer.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: peviewer.asm

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

图 4.4: 编译

4.4.2 链接



```
E:\masm_code\lab4>D:\masm32\bin\Link /SUBSYSTEM:CONSOLE peviewer.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

图 4.5: 链接

4.4.3 执行

```
E:\masm_code\lab4>. \peviewer.exe
Please input a PE file :hello_console.exe

IMAGE_DOS_HEADER
  e_magic: 5A4D
  e_lfanew: 000000B0
IMAGE_NT_HEADERS
  Signature: 00004550
IMAGE_FILE_HEADER
  NumberOfSections: 0003
  TimeDateStamp: 6528E409
  Characteristics: 010F
IMAGE_OPTIONAL_HEADER
  AddressOfEntryPoint: 00001000
  ImageBase: 00400000
  SectionAlignment: 00001000
  FileAlignment: 00000200
E:\masm_code\lab4>
```

图 4.6: 运行结果