

南开大学

汇编语言与逆向技术课程实验报告

实验五：PEViewer



学 院 网络空间安全学院
专 业 信息安全
学 号 2211044
姓 名 陆皓喆
班 级 信息安全

一、程序的设计说明和控制流程图

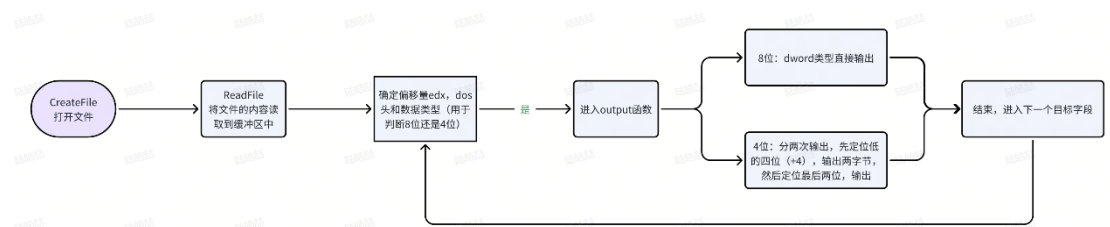
1.1 程序设计说明

该程序的主要流程是利用已给出的打开文件的函数，调用之前写过的一些.exe 文件，然后再利用 `edx` 作为偏移量，存储寄存器，根据 PE 文件中已知的偏移量，每次对 `edx` 进行 `add` 操作，便于访问到每一个需要输出的字段的值。

代码中的 `Output` 过程是用来输出各个字段的值，其中利用 `ecx` 来计数判断应输出当前位置还是下一位置的字段值。在该过程中还调用了之前写过的 `dw2hex` 函数，将值转化为 16 进制并输出。

1.2 程序控制流程图

程序主要的控制流程图如下所示：



二、源代码以及注释

```
.386
.386
.model flat,stdcall
option casemap :none
include \masm32\include\windows.inc
include \masm32\include\masm32.inc
include \masm32\macros\macros.asm
include \masm32\include\kernel32.inc
includelib \masm32\lib\masm32.lib
includelib \masm32\lib\kernel32.lib

.data
;数据段
;定义一些输出文本数据
que BYTE "Please input a PE file: ",0
que1 BYTE "IMAGE_DOS_HEADER",0Ah,0Dh,0
que2 BYTE "    e_magic:",0
que3 BYTE "    e_lfanew:",0
```

```

que4 BYTE "IMAGE_DOS_HEADER",0Ah,0Dh,0
que5 BYTE "    Signature:",0
que6 BYTE "IMAGE_DOS_HEADER",0Ah,0Dh,0
que7 BYTE "    NumberOfSections:",0
que8 BYTE "    TimeDateStamp:",0
que9 BYTE "    Charateristics:",0
que10 BYTE "IMAGE_DOS_HEADER",0Ah,0Dh,0
que11 BYTE "    e_magic:",0
que12 BYTE "    e_lfanew:",0
que13 BYTE "IMAGE_DOS_HEADER",0Ah,0Dh,0
que14 BYTE "    AddressOfEntryPoint:",0
que15 BYTE "    ImageBase:",0
que16 BYTE "    SectionAlignment:",0
que17 BYTE "    FileAlignment:",0
;对上面的几十句话进行后续的分别输出
buf3 DWORD 4000 DUP(0)
;缓冲区指针（必须把空间开的足够大）
buf4 DWORD 4000 DUP(0)
;文件内容，转成十六进制存储
buf5 WORD 4000 DUP(0)
;存储后四位
file BYTE 20 DUP(0),0
;文件名
hfile DWORD 0,0
;文件句柄

endl BYTE 0Ah,0Dh,0
;换行
temp DWORD 0,0
;存 ecx 的值（是四位还是八位）
temp1 DWORD 0,0
;定位指针的初始位置，是 Dos 头还是 PE 头
;定义代码段
.code
Output PROC
;读取相应位置的内容并转化成二进制
;edx 为偏移量
    mov esi,OFFSET buf3
    add esi,edx
;把偏移量加上
    add esi,temp1
;DOS 头还是 NT 头
    mov eax,DWORD PTR[esi]
;将最终定位的地址赋给 eax 寄存器

```

```

        mov ebx,eax
        invoke dw2hex,eax,addr buf4
;将 eax 转换为 16 进制后存入 buf4 中
        mov ecx,temp
        .if ecx==8
            invoke StdOut,addr buf4
;如果查表结果是 dword 型，直接输出
        .else;不是 dword 型，就是四位，按照分两次的方法进行输出
;如果查表结果是 word 型
            mov ax,WORD PTR [buf4+4]
;先定位到低四位
            mov buf5,ax
            invoke StdOut,addr buf5
;输出两个字节

```

```

        mov ax,WORD PTR [buf4+6]
;再定位到后两位
        mov buf5,ax
        invoke StdOut,addr buf5
;再输出两个字节
        .endif
        invoke StdOut,addr endl
ret
Output ENDP

```

```

start:
        invoke StdOut,addr que
;输出请求
        invoke StdIn,addr file,20
;输入 exe 名称
;打开文件
        invoke CreateFile,addr file,\
                                GENERIC_READ,\
                                FILE_SHARE_READ,\
                                0,\
                                OPEN_EXISTING,\
                                FILE_ATTRIBUTE_ARCHIVE,\
                                0

        mov hfile,eax
;保存文件句柄
        invoke SetFilePointer,hfile,0,0,FILE_BEGIN
        invoke ReadFile,hfile,addr buf3,4000,0,0
        mov esi,OFFSET buf3
;文件入口

```

```
    invoke StdOut,addr que1
    invoke StdOut,addr que2
;下面就是根据表格定位指针，然后带入 output 函数就行，都是一个意思
;EM
```

```
    mov edx,0
    mov temp1,edx
    mov ecx,4
    mov temp,ecx
    invoke Output
```

```
    invoke StdOut,addr que3
;EF
```

```
    mov edx,3ch
    mov ecx,8
    mov temp,ecx
    invoke Output
```

```
    invoke StdOut,addr que4
    invoke StdOut,addr que5
    mov temp1,ebx
;S
```

```
    mov edx,0
    invoke Output
```

```
    invoke StdOut,addr que6
    invoke StdOut,addr que7
;NOS
```

```
    mov edx,6h
    mov ecx,4
    mov temp,ecx
    invoke Output
```

```
    invoke StdOut,addr que8
;TDS
```

```
    mov edx,8h
    mov ecx,8
    mov temp,ecx
    invoke Output
```

```
    invoke StdOut,addr que9
;C
```

```
    mov edx,16h
    mov ecx,4
    mov temp,ecx
```

```

        invoke Output

        invoke StdOut,addr que13
        invoke StdOut,addr que14
;AOE
        mov edx,28h
        mov ecx,8
        mov temp,ecx
        invoke Output

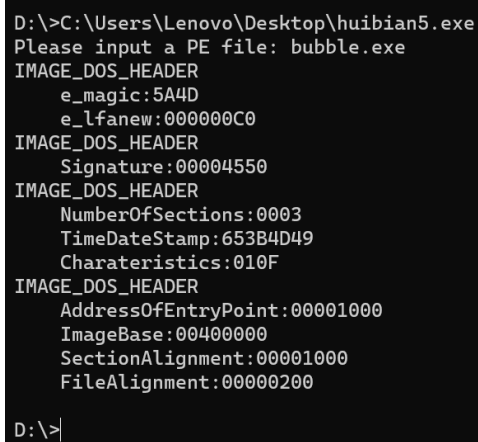
        invoke StdOut,addr que15
;IB
        mov edx,34h
        invoke Output

        invoke StdOut,addr que16
;SA
        mov edx,38h
        invoke Output

        invoke StdOut,addr que17
;FA
        mov edx,3ch
        invoke Output
        invoke CloseHandle,hfile
        invoke ExitProcess,0
end start

```

三、运行截图



```

D:\>C:\Users\Lenovo\Desktop\huibian5.exe
Please input a PE file: bubble.exe
IMAGE_DOS_HEADER
e_magic:5A4D
e_lfanew:000000C0
IMAGE_DOS_HEADER
Signature:00004550
IMAGE_DOS_HEADER
NumberOfSections:0003
TimeDateStamp:653B4D49
Charateristics:010F
IMAGE_DOS_HEADER
AddressOfEntryPoint:00001000
ImageBase:00400000
SectionAlignment:00001000
FileAlignment:00000200
D:\>

```

上面的图片是对之前几次实验中的 bubble 实验进行了验证，发现均符合预期。