**实验4 peviewer**

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. **实验原理**

**（1）PE文件结构**

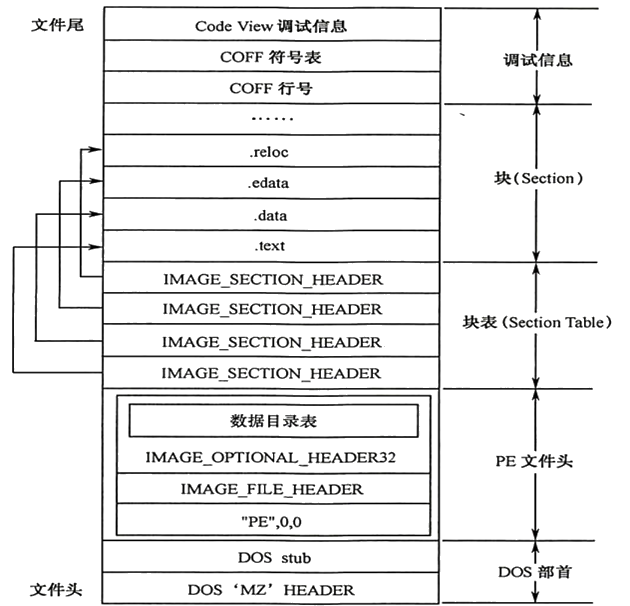


图 1 PE文件结构

PE文件结构如图1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。

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。

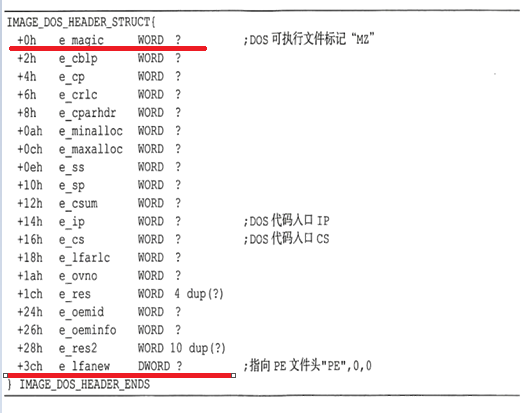


图 2 DOS头的数据结构

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（EntryPoint）、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：

IMAGE\_NT\_HEADERS STRUCT

+0h Signature DWORD ?

+4h FileHeader IMAGE\_FILE\_HEADER <>

+18h OptionalHeader IMAGE\_OPTIONAL\_HEADER32 <>

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：

IMAGE\_FILE\_HEADER STRUCT

+04h Machine WORD ?

+06h NumberOfSections WORD ?

+08h TimeDateStamp DWORD ？

+0Ch PointerToSymbolTable DWORD ?

+10h NumberOfSymbols DWORD ?

+14h SizeOfOptionalHeader WORD ？

+16h Characteristics WORD ?

**OptionalHeader**的数据结构如下所示：

IMAGE\_OPTIONAL\_HEADER  STRUCT

+18h Magic WORD ？

+1Ah MajorLinkerVersion BYTE ？

+1Bh MinorLinkerVersion BYTE ？

+1Ch SizeOfCode DWORD ？

+20h SizeOfInitializedData DWORD ？

+24h SizeOfUninitializedData DWORD ？

+28h AddressOfEntryPoint DWORD ？

+2Ch BaseOfCode DWORD ？

+30h BaseOfData DWORD ？

+34h ImageBase DWORD ？

+38h SectionAlignment DWORD  ？

+3Ch FileAlignment DWORD  ？

+40h MajorOperatingSystemVersion WORD ？

+42h MinorOperatingSystemVersion WORD ？

+44h MajorImageVersion WORD ？

+46h MinorImageVersion WORD ？

+48h MajorSubsystemVersion WORD ？

+4Ah MinorSubsystemVersion WORD ？

+4Ch Win32VersionValue DWORD ？

+50h SizeOfImage DWORD ？

+54h SizeOfHeaders DWORD ？

+58h CheckSum DWORD ？

+5Ch Subsystem WORD ？

+5Eh DllCharacteristics WORD ？

+60h SizeOfStackReserve DWORD ？

+64h SizeOfStackCommit DWORD ？

+68h SizeOfHeapReserve DWORD ？

+6Ch SizeOfHeapCommit DWORD ？

+70h LoaderFlags DWORD ？

+74h NumberOfRvaAndSizes DWORD ？

+78h DataDirectory [IMAGE\_NUMBEROF\_DIRECTORY\_ENTRIES] IMAGE\_DATA\_DIRECTORY <>

IMAGE\_OPTIONAL\_HEADER ENDS

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。

**（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**的MSDN文档地址<https://docs.microsoft.com/en-us/previous-versions/aa914735(v=msdn.10)>，函数的原型如下：

HANDLE CreateFile(

LPCTSTR lpFileName,

DWORD dwDesiredAccess,

DWORD dwShareMode,

LPSECURITY\_ATTRIBUTES lpSecurityAttributes,

DWORD dwCreationDisposition,

DWORD dwFlagsAndAttributes,

HANDLE hTemplateFile

);

CreateFile在MASM汇编语言中的应用实例如图3所示。

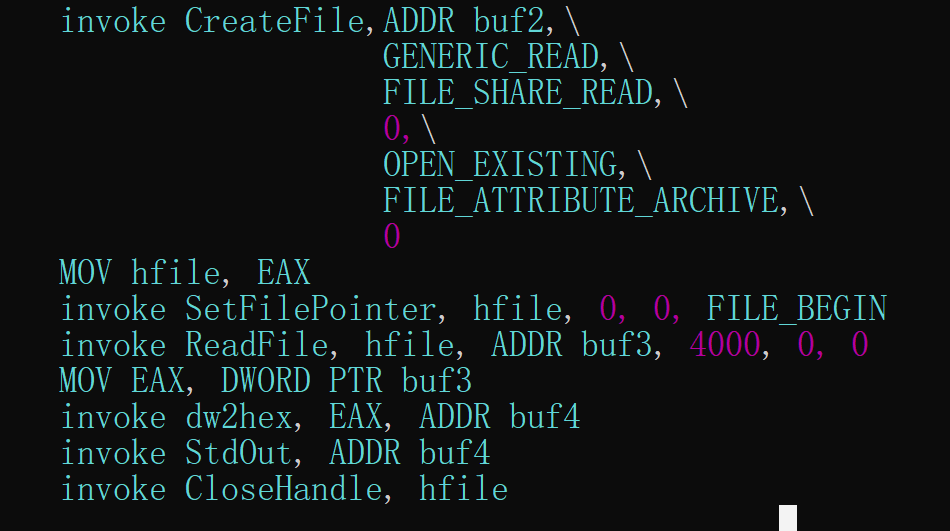


图 3 MASM汇编中调用CreateFile、SetFilePointer、ReadFile、CloseHandle的示例

**SetFilePointer**函数的MSDN文档地址 <https://docs.microsoft.com/en-us/previous-versions/aa911934(v%3dmsdn.10)>，函数原型如下：

DWORD SetFilePointer(

HANDLE hFile,

LONG lDistanceToMove,

PLONG lpDistanceToMoveHigh,

DWORD dwMoveMethod

);

SetFilePointer函数的MASM调用示例如图3所示。

**ReadFile**函数的MSDN文档地址<https://docs.microsoft.com/en-us/previous-versions/aa914377(v%3dmsdn.10)>，函数原型如下：

BOOL ReadFile(

HANDLE hFile,

LPVOID lpBuffer,

DWORD nNumberOfBytesToRead,

LPDWORD lpNumberOfBytesRead,

LPOVERLAPPED lpOverlapped

);

ReadFile函数的MASM调用示例如图3所示。

**CloseHandle**函数的MSDN文档地址<https://docs.microsoft.com/en-us/previous-versions/aa914720(v%3dmsdn.10)>，函数原型如下：

BOOL CloseHandle(

HANDLE hObject

);

CloseHandle函数的MASM调用示例如图3所示。

1. **实验环境**

Windows操作系统，MASM32编译环境。

1. **实验内容：**

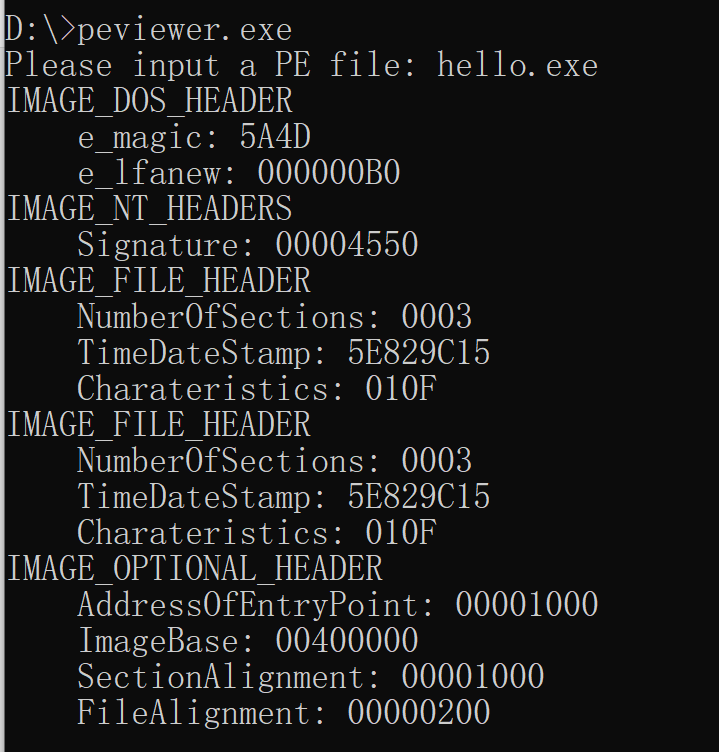


图 4 peviewer实验演示

* + 1. 输入PE文件的文件名，peviewer程序调用Windows API函数，打开指定的PE文件；
    2. 从文件的头部开始，读取IMAGE\_DOS\_HEADER结构中的e\_magic和e\_lfanew字段的值，按照实验演示的方式输出到命令行窗口；
    3. 继续读取PE文件的IMAGE\_NT\_HEADER结构中的Signature字段的值，按照实验演示的方式输出到命令行窗口；
    4. 继续读取IMAGE\_NT\_HEADER结构中的IMAGE\_FILE\_HEADER结构，从中读取出字段NumberOfSections、TimeDateStamp、Characteristics的值，按照实验演示的方式输出到命令行窗口；
    5. 继续读取IMAGE\_NT\_HEADER结构中的IMAGE\_OPTIONAL\_HEADER结构，从中读取字段AddressOfEntryPoint、ImageBase、SectionAlignment、FileAlignment的值，按照实验演示的方式输出到命令行窗口；

1. **实验报告**
   * 1. peviewer程序的设计说明和控制流图
     2. peviewer.asm的源代码和注释
     3. peviewer.exe运行截图