test.exe 程序-PE 文件格式分析

姓名：_____　　学号：_____

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000000h: 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 ; MZ?.....  ..
00000010h: B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ; ?......@.......
00000020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000030h: 00 00 00 00 00 00 00 00 00 00 00 00 B0 00 00 00 ; ............?...
00000040h: 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ; ..?.???L?Th
00000050h: 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F ; is program canno
00000060h: 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 ; t be run in DOS
00000070h: 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 ; mode....$.......
00000080h: 5D 65 FD C8 19 04 93 9B 19 04 93 9B 19 04 93 9B ; ]e  ..搪..搪..搪
00000090h: 97 1B 80 9B 11 04 93 9B E5 24 81 9B 18 04 93 9B ; ?€?.搪?仜..搪
000000a0h: 52 69 63 68 19 04 93 9B 00 00 00 00 00 00 00 00 ; Rich..搪........
```

开始位置　字串　x86　3 个节　生成时间　COFF 表偏移

```
000000b0h: 50 45 00 00 4C 01 03 00 9B 4D 8F 42 00 00 00 00 ; PE..L...汳廋....
```

符号数目　可选头大小　标记　幻数　主次版本　代码段总尺寸

```
000000c0h: 00 00 00 00 E0 00 0F 01 0B 01 05 0C 00 02 00 00 ; ....?.........
```

已初始数据尺寸　未初始数据尺寸　代码节开始位置

```
000000d0h: 00 04 00 00 00 00 00 00 00 10 00 00 00 10 00 00 ; ................
```

Base of Data　Image Base

数据节开始位置　默认装载地址，一般是 400000H

```
000000e0h: 00 20 00 00 00 00 40 00 00 10 00 00 00 02 00 00 ; . ....@.........
000000f0h: 04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 ; ................
00000100h: 00 40 00 00 00 04 00 00 00 00 00 00 02 00 00 00 ; .@..............
00000110h: 00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00 ; ................
00000120h: 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000130h: 14 20 00 00 3C 00 00 00 00 00 00 00 00 00 00 00 ; . ..<...........
00000140h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000150h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000160h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000170h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000180h: 00 00 00 00 00 00 00 00 00 20 00 00 14 00 00 00 ; ......... ......
00000190h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

节名 8 字节

```
000001a0h: 00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00 ; ........text...
```

OBJ：本节物理地址

EXE:节的实际字节数　相对虚拟地址　本节对齐后尺寸　本节原始数据在文件中位置

```
000001b0h: 46 00 00 00 00 10 00 00 00 02 00 00 00 04 00 00 ; F...............
```

OBJ：该节重定位偏移　重定位 行号表

EXE:无意义　行号偏移　数目　行号数　节属性

```
000001c0h: 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60 ; ............ ..`
000001d0h: 2E 72 64 61 74 61 00 00 A6 00 00 00 00 20 00 00 ; .rdata..?... ..
000001e0h: 00 02 00 00 00 06 00 00 00 00 00 00 00 00 00 00 ; ................
000001f0h: 00 00 00 00 40 00 00 40 2E 64 61 74 61 00 00 00 ; ....@..@.data...
00000200h: 8E 00 00 00 00 30 00 00 00 02 00 00 00 08 00 00 ; ?...0...........
00000210h: 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0 ; ............@..?
00000220h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000230h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000240h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000250h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000260h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000270h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000280h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000290h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000002a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000002b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000002c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000002d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000002e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000002f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000300h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000310h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000320h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000330h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000340h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000350h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000360h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000370h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000380h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000390h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000003f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

```
00000400h: 68 40 10 00 00 68 00 30 40 00 68 09 30 40 00 6A ; h@...h.0@.h.0@.j
00000410h: 00 E8 2A 00 00 00 68 40 10 00 00 68 00 30 40 00 ; .?...h@...h.0@.
00000420h: 68 31 30 40 00 6A 00 E8 14 00 00 00 6A 00 E8 01 ; h10@.j.?...j.?
00000430h: 00 00 00 CC FF 25 00 20 40 00 FF 25 0C 20 40 00 ; ...?%. @.  %. @.
00000440h: FF 25 08 20 40 00 00 00 00 00 00 00 00 00 00 00 ;  %. @...........
00000450h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000460h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000470h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000480h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000490h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000004f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000500h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000510h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000520h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000530h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000540h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000550h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000560h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000570h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000580h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

MZ 文件头，默认 0x40 字节，开头字符为 MZ

DOS 桩小程序，用于在 Dos 下提示无法运行的信息。信息提示不唯一

字串”PE\0\0”4 字节

映像文件头，14H 字节

可选文件头，00E0H 字节

(由映像文件头字段决定

三者共同组成 PE 文件头

PE 文件头

DIRECTORY:数据目录是一个 IMAGE_DATA_DIR 数组，存放着 PE 一些重要部分的起始 RVA 和尺寸。使得更快装载。

16 项 * 8 字节 = 128 字节

每项：

VirtualAddress :双字

Size : 双字

最后四字节 3CH-3FH，定位 PE 文件头开始位置，可用于检测 PE 文件合法性

Dos 程序的提示信息

Address of entry point

00001000H

开始执行位置(第一条代码的 RVA，

Base of Code

SectionAlignment，PE 文件转入内存节对齐数字，1000H=4K

FileAligment，文件节对齐数字，200H=516Bytes

节表，每项 28H 字节，包含一个节的具体信息。

节表项个数由映像文件头 02H-04H 决定，可知 0003H，本 PE 文件有三个节

文件中节对齐力度是 200H，用 00 对齐

节对齐，到节表部分为止 PE 文件共占据 200H+200H=400H

代码节 .test / .CODE

实际大小 46H(图中，不同系统不同)，对齐大小 200H

包含 PE 文件的可执行代码，一般的 PE 文件都会包含代码节

代码节各个字段

内存中的代码节

```
$  68 40100000    push 0x1040
.  68 00304000    push test.00403000
.  68 09304000    push test.00403009
.  6A 00          push 0x0
.  E8 24000000    call <jmp.&user32.MessageBoxA>
.  68 40100000    push 0x1040
.  68 00304000    push test.00403000
.  68 30304000    push test.00403030
.  6A 00          push 0x0
.  E8 0E000000    call <jmp.&user32.MessageBoxA>
.  6A 00          push 0x0
.  E8 01000000    call <jmp.&kernel32.ExitProcess>
   CC             int3
.- FF25 0020400    jmp dword ptr ds:[<&kernel32.ExitProces
$- FF25 0820400    jmp dword ptr ds:[<&user32.MessageBoxA>
```

14 20 00 00：RVA 地址

8C 00 00 00：IDT 表的大小

IDT 表

虽然 OriginalFirstThunk 和 FirstThunk 在文件中指向的字段数据相同,但是在内存中FirstThunk 指向的值会被函数在内存中对应的真实地址代替（动态链接机制

INT 引入名字表

pFile	Data	Description
00000650	00002064	Hint/Name RVA
00000654	00000000	End of Imports
00000658	0000208C	Hint/Name RVA
0000065C	00002080	Hint/Name RVA
00000660	00000000	End of Imports

Data 最高位：

为 0，表示通过函数名引入，指向 Impotr Hints/Names

为1，表示通过序号引入函数

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
-------------------------------------------------------------------
00000590h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000005f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
                    IAT 表
00000600h: 64 20 00 00 00 00 00 00 8C 20 00 00 80 20 00 00 ; d ......?.. €  ..
                    OriginalFirstThunk
00000610h: 00 00 00 00 50 20 00 00 00 00 00 00 00 00 00 00 ; ....P ..........
                    FirstThunk
00000620h: 72 20 00 00 00 20 00 00 58 20 00 00 00 00 00 00 ; r ... ..X .....
00000630h: 00 00 00 00 9A 20 00 00 08 20 00 00 00 00 00 00 ; ....?.. ........
00000640h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
                    Imoprt Name Table 引入名字表
00000650h: 64 20 00 00 00 00 00 00 8C 20 00 00 80 20 00 00 ; d ......?.. €  ..
                    Import HINTS/Names & DLL Names
                    Hints   引入函数名
00000660h: 00 00 00 00 00 00 45 78 69 74 50 72 6F 63 65 73 ; ......€.ExitProces
00000670h: 73 00 6B 65 72 6E 65 6C 33 32 2E 64 6C 6C 00 00 ; s.kernel32.dll..
                    Hints                   Hints
00000680h: 00 00 77 73 70 72 69 6E 74 66 41 00 00 00 4D 65 ; ..wsprintfA...Me
00000690h: 73 73 61 67 65 42 6F 78 41 00 75 73 65 72 33 32 ; ssageBoxA.user32
000006a0h: 2E 64 6C 6C 00 00 00 00 00 00 00 00 00 00 00 00 ; .dll............
000006b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000006c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000006d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000006e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000006f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000700h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000710h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000720h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000730h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000740h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000750h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000760h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000770h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000780h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000790h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000007f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
-------------------------------------------------------------------

00000800h: BD CC D1 A7 B2 E2 CA D4 00 50 45 C8 EB BF DA B5 ; 教学测试.PE 入口?
00000810h: E3 B2 E2 CA D4 31 A3 BA BD F8 C8 EB B5 DA D2 BB ; 悴馐?：进入第一
00000820h: C8 EB BF DA CE BB D6 C3 34 30 31 30 30 30 48 21 ; 入口位置 401000H!
00000830h: 00 50 45 C8 EB BF DA B5 E3 B2 E2 CA D4 32 A3 BA ; .PE 入口点测试 2：
00000840h: BD F8 C8 EB B5 DA B6 FE C8 EB BF DA CE BB D6 C3 ; 进入第二入口位置
00000850h: 34 30 31 30 31 36 48 21 00 00 00 00 00 00 00 00 ; 401016H!........
00000860h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000870h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

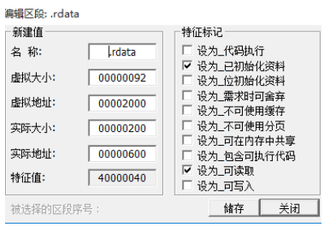
内存中的数据节

Address	Hex dump	ASCII
00403000	BD CC D1 A7 B2 E2 CA D4 00 50 45 C8 EB BF DA B5	教学测试.PE入口?
00403010	E3 B2 E2 CA D4 31 3A BD F8 C8 EB B5 DA D2 BB C8	悴馐?:进入第一?
00403020	EB BF DA CE BB D6 C3 34 30 31 30 30 30 48 21 00	肪谖恢?01000H!.
00403030	50 45 C8 EB BF DA B5 E3 B2 E2 CA D4 32 3A BD F8	PE入口点测试2:进
00403040	C8 EB B5 DA D2 BB C8 EB BF DA CE BB D6 C3 34 30	入第一入口位置40
00403050	31 30 31 36 48 21 00 00 00 00 00 00 00 00 00 00	1016H!..........
00403060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	----------------
00403070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	----------------
00403080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	----------------

引入函数节 .idata/rdata

实际大小 92H，对齐大小 200H

引入函数节字段特征



IDT 表的 IMAGE_IMPORT_DESCRIPTOR 结构元素 ：

数据节 .data (已初始化)

实际大小 57H，对齐大小 200H

存放在编译时刻就已经确定好的数据

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
-------------------------------------------------------------------
00000880h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000890h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000008f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000900h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000910h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000920h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000930h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000940h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000950h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000960h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000970h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000980h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
00000990h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009b0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009d0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
000009f0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ................
```

代码节各个字段和特征

要求：

1. 分割 PE 文件的各个部分：MZ 头部+DOS Stub+PE 文件头+可选文件头+节表+节
2. 标明各个关键字段结构和字段，及其含义
3. 详细分析函数引入表中的各个字段及其关系

pFile	Data	Description	Value	
00000614	00002050	Import Name Table RVA		
00000618	00000000	Time Date Stamp		
0000061C	00000000	Forwarder Chain		
00000620	00002072	Name RVA	kernel32.dll	IMAGE_IMPORT_DESCRIPTOR
00000624	00002000	Import Address Table RVA		
00000628	00002058	Import Name Table RVA		
0000062C	00000000	Time Date Stamp		
00000630	00000000	Forwarder Chain		
00000634	0000209A	Name RVA	user32.dll	IMAGE_IMPORT_DESCRIPTOR
00000638	00002008	Import Address Table RVA		
0000063C	00000000			
00000640	00000000			
00000644	00000000			
00000648	00000000			
0000064C	00000000			IMAGE_IMPORT_DESCRIPTOR（最后一项全0