

编号: _____

实验	一	二	三	四	五	六	七	八	总评	教师签名
成绩										

武汉大学国家网络安全学院

课程实验(设计)报告

题 目: _____ 作业 1: 最小 PE 压缩

专业(班): _____ 2021 级信安 6 班

学 号: _____ 2021302181156

姓 名: _____ 赵伯侯

课程名称: _____ 软件安全实验

任课教师: _____ 赵磊

2030 年 12 月 4 日

目 录

1 实验名称	1
2 实验目的	1
3 实验步骤及内容	1
(1) 删除 DOS 头和 DOS_STUB 结构中不重要数据。	1
(2) 修改可选文件头的数据目录	2
(3) 修改节表内容。	2
(4) 修改代码节的内容。	4
(5) 修改函数调用内容。	4
(6) 只读数据段重构。	5
(7) 补充函数代码。	6
(8) 修改参数	7
4.结果展示	8

1 实验名称

最小 PE 压缩

2 实验目的

自己打造一个尽可能小的 PE 文件

3 实验步骤及内容

(1) 删除 DOS 头和 DOS_STUB 结构中不重要数据。

打开文件的 DOS 头部分将不重要的部分标记为 1 后如下图所示。

0000h:	4D	5A	11	11	11	11	11	11	11	11	11	11	11	11	11	MZ.....
0010h:	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
0020h:	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
0030h:	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11

DOS 头文件中重要的数据仅有两个，其一是 DOS 的签名占有两个字节，其二是 DOS 头的末尾 003C 位置的四个字节，指向 NT 头的文件内偏移。其余的数据都可以删去。

同时 DOS_STUB 结构中的所有数据都是不重要的数据可以全部删除，将二者完全删除后的结果如下图所示。

0000h:	4D	5A	11	11	50	45	00	00	4C	01	02	00	04	FE	24	45	MZ..PE..L...p\$E
0010h:	00	00	00	00	00	00	00	00	E0	00	0F	01	0B	01	05	0Cà.....
0020h:	74	00	00	00	A8	00	00	00	00	00	00	00	3A	02	00	00	t...".....
0030h:	F8	01	00	00	6C	02	00	00	00	00	40	00	04	00	00	00	ø...l.....@.....
0040h:	04	00	00	00	04	00	00	00	00	00	00	00	04	00	00	00
0050h:	00	00	00	00	14	03	00	00	F8	01	00	00	00	00	00	00ø.....
0060h:	02	00	00	00	00	00	10	00	00	10	00	00	00	00	10	00
0070h:	00	10	00	00	00	00	00	00	10	00	00	00	00	00	00	00
0080h:	00	00	00	00	80	02	00	00	3C	00	00	00	00	00	00	00€...<.....
0090h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A0h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

保留 DOS 签名后的两个字节使得 NT 头的位置为 0004H 方便寻址与粒度的配合。这里 003C 位置的数据内容是 00000004，代表 SectionAlignment 指定了节段在内存中的最小单位为 4，指向 NT 文件头，正好是 PE 文件头的位置。

(2) 修改可选文件头的数据目录

由于数据目录一共有 16 项，但在该文件中只有导入表会被使用所以将导入表即数组第二项之后的内容全部删除，并且将变量 `NumberOfRvaAndSizes` 的值改为 00000002，代表数据目录只有两项。即将下图中选中的部分进行删除。

0060h:	02 00 00 00	00 00 10 00	00 10 00 00	00 00 10 00
0070h:	00 10 00 00	00 00 00 00	10 00 00 00	00 00 00 00
0080h:	00 00 00 00	80 02 00 00	3C 00 00 00	00 00 00 00
0090h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00A0h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00B0h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00C0h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00D0h:	00 00 00 00	00 00 00 00	00 00 00 00	6C 02 00 00
00E0h:	14 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
00F0h:	00 00 00 00	00 00 00 00	00 00 00 00	2E 74 65 78
0100h:	74 00 00 00	72 00 00 00	F8 01 00 00	74 00 00 00
0110h:	F8 01 00 00	00 00 00 00	00 00 00 00	00 00 00 00

修改完成后的结果如下图所示。

0000h:	4D 5A 14 14	50 45 00 00	4C 01 02 00	04 FE 24 45	MZ...PE...L....pSE
0010h:	00 00 00 00	00 00 00 00	E0 00 0F 01	0B 01 05 0Cä.....
0020h:	74 00 00 00	A8 00 00 00	00 00 00 00	3A 02 00 00	t.....@.....
0030h:	F8 01 00 00	6C 02 00 00	00 00 40 00	04 00 00 00	ø...l.....@.....
0040h:	04 00 00 00	04 00 00 00	00 00 00 00	04 00 00 00ø.....
0050h:	00 00 00 00	14 03 00 00	F8 01 00 00	00 00 00 00ø.....
0060h:	02 00 00 00	00 00 10 00	00 10 00 00	00 00 10 00ø.....
0070h:	00 10 00 00	00 00 00 00	02 00 00 00	00 00 00 00ø.....
0080h:	00 00 00 00	80 02 00 00	BC 00 00 00	2E 74 65 78ø...<....tex
0090h:	74 00 00 00	72 00 00 00	F8 01 00 00	74 00 00 00	t...r...ø...t...
00A0h:	F8 01 00 00	00 00 00 00	00 00 00 00	00 00 00 00	ø.....@.....
00B0h:	20 00 00 60	14 14 64 61	74 61 00 00	A6 00 00 00	...rddata...!
00C0h:	6C 02 00 00	A8 00 00 00	6C 02 00 00	00 00 00 00	l...".l.....
00D0h:	00 00 00 00	00 00 00 00	40 00 00 40	4D 79 6D 69@...@Mymi
00E0h:	6E 69 45 58	45 2C 73 69	7A 65 3A 37	38 38 42 00	niEXE,size:788B.
00F0h:	CE E4 B4 F3	D0 C5 B0 B2	50 45 D7 F7	D2 B5 28 D0	Ia'6DÄ"2PE×+Op(D
0100h:	D5 C3 FB 3A	C4 B3 C4 B3	C4 B3 2C D1	A7 BA C5 3A	ÖÄü:A"A"A',Ns:A:
0110h:	32 30 30 33	33 32 31 31	30 2A 2A 2A	29 00 68 40	200332110***)..h@
0120h:	10 00 00 68	F8 01 40 00	68 0C 02 40	00 6A 00 E8	...hø.@.h...@.j.e
0130h:	14 00 00 00	3A 00 00 00	00 00 00 CC	FF 25 6C 02	...j...i%l.
0140h:	40 00 FF 25	78 02 40 00	FF 25 74 02	40 00 00 00	@.y%x.@.y%t.@...
0150h:	00 00 00 00	00 00 00 00	F8 02 00 00	EC 02 00 00ø...i...
0160h:	00 00 00 00	BC 02 00 00	00 00 00 00	00 00 00 00k.....
0170h:	DE 02 00 00	6C 02 00 00	C4 02 00 00	00 00 00 00	P...l...Ä.....
0180h:	00 00 00 00	06 03 00 00	74 02 00 00	00 00 00 00k.....
0190h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00ø...i...
01A0h:	D0 02 00 00	00 00 00 00	F8 02 00 00	EC 02 00 00	D.....ø...i...
01B0h:	00 00 00 00	80 00 45 78	69 74 50 72	6F 63 65 73ø.ExitProces
01C0h:	73 00 6B 65	72 6E 65 6C	33 32 2E 64	6C 6C 00 00	s.kernel32.dll...
01D0h:	62 02 77 73	70 72 69 6E	74 66 41 00	9D 01 4D 65	b.wsprintfA...Me
01E0h:	73 73 61 67	65 42 6F 78	41 00 75 73	65 72 33 32	ssageBoxA.user32
01F0h:	2E 64 6C 6C	00 00 00 00			.dll....

(3) 修改节表内容。

接下来修改文件的节表相关的内容，打开可执行文件的节表如下图所示。

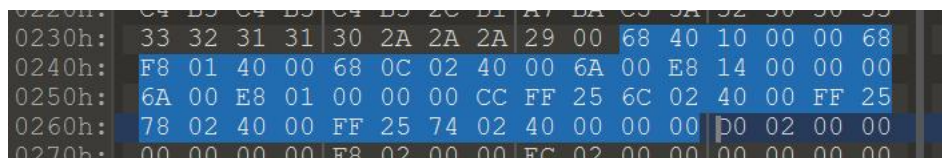
0080h:	00 00 00 00	80 02 00 00	3C 00 00 00	2E 74 65 78ø...<....tex
0090h:	74 00 00 00	72 00 00 00	F8 01 00 00	74 00 00 00	t...r...ø...t...
00A0h:	F8 01 00 00	00 00 00 00	00 00 00 00	00 00 00 00	ø.....@.....
00B0h:	20 00 00 60	2E 72 64 61	74 61 00 00	A6 00 00 00	...rddata...!
00C0h:	6C 02 00 00	A8 00 00 00	6C 02 00 00	00 00 00 00	l...".l.....
00D0h:	00 00 00 00	00 00 00 00	40 00 00 40	4D 79 6D 69@...@Mymi
00E0h:	6E 69 45 58	45 2C 73 69	7A 65 3A 37	38 38 42 00	niEXE,size:788B.

在本次实验中需要保留代码节的相关部分，将只读数据节的部分即上图中红框中的部分进行删除。

(4) 修改代码节的内容。

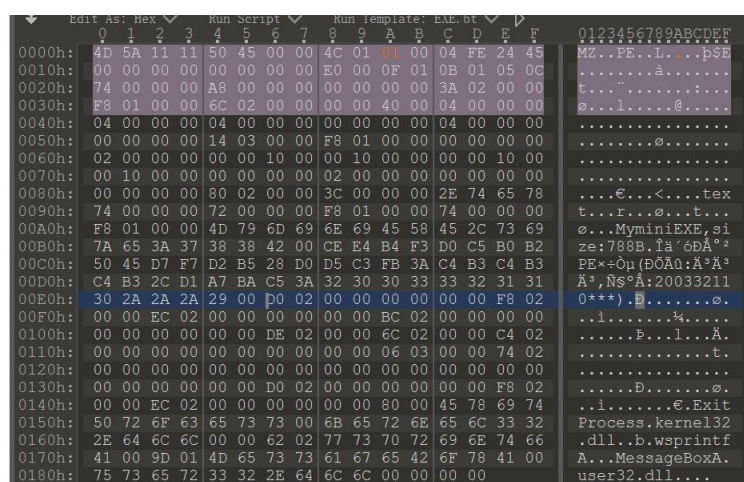
由于需要尽可能将文件的大小压至最小，所以代码部分需要进行修改。

首先在尚未修改的 PE 文件中找到可选文件头中 AddressOfEntryPoint 变量指向的程序代码的起始部分为 023A 然后在原 PE 文件中查找目录项中的 ImportAddressTable 指向的位置为 026C 由此可以得到整个代码段的位置如下图所示。



0230h:	33 32 31 31	30 2A 2A 2A	29 00 68 40	10 00 00 68
0240h:	F8 01 40 00	68 0C 02 40	00 6A 00 E8	14 00 00 00
0250h:	6A 00 E8 01	00 00 00 CC	FF 25 6C 02	40 00 FF 25
0260h:	78 02 40 00	FF 25 74 02	40 00 00 00	D0 02 00 00
0270h:	00 00 00 00	F8 02 00 00	EC 02 00 00	00 00 00 00

将当前的代码段删除方便以后的步骤中重新编写新的代码段。删除后的结果如下图所示。



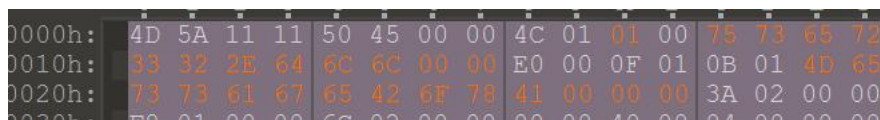
0000h:	4D 5A 11 11	50 45 00 00	4C 01 01 00	04 FE 24 45	MZ..PE...L...p\$E
0010h:	00 00 00 00	00 00 00 00	E0 00 0F 01	0B 01 05 0CA.....
0020h:	74 00 00 00	A8 00 00 00	00 00 00 00	3A 02 00 00	t..."......
0030h:	F8 01 00 00	6C 02 00 00	00 00 40 00	04 00 00 00l.....e.....
0040h:	04 00 00 00	04 00 00 00	00 00 00 00	04 00 00 00
0050h:	00 00 00 00	14 03 00 00	F8 01 00 00	00 00 00 00e.....
0060h:	02 00 00 00	00 00 10 00	00 10 00 00	00 00 10 00
0070h:	00 10 00 00	00 00 00 00	02 00 00 00	00 00 00 00
0080h:	00 00 00 00	80 02 00 00	3C 00 00 00	2E 74 65 78e...<...tex
0090h:	74 00 00 00	72 00 00 00	F8 01 00 00	74 00 00 00	t...r...e...t...
00A0h:	F8 01 00 00	4D 79 6D 69	6E 69 45 58	45 2C 73 69	ø...MyMiniEXE,si
00B0h:	7A 65 3A 37	38 38 42 00	CE E4 B4 F3	D0 C5 B0 B2	ze:788B.fâ'6dA'^2
00C0h:	50 45 D7 F7	D2 B5 28 D0	D5 C3 FB 3A	C4 B3 C4 B3	PE*+Op(D0A0:A^A^
00D0h:	C4 B3 2C D1	A7 BA C5 3A	32 30 30 33	33 32 31 31	A^,Ns^A:20033211
00E0h:	30 2A 2A 2A	29 00 p0 02	00 00 00 00	00 00 F8 02	(0**).B.....ø.
00F0h:	00 00 EC 02	00 00 00 00	00 00 BC 02	00 00 00 00	..i.....%.....
0100h:	00 00 00 00	00 00 DE 02	00 00 6C 02	00 00 C4 02P...l...A.
0110h:	00 00 00 00	00 00 00 00	00 00 06 03	00 00 74 02t.....
0120h:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0130h:	00 00 00 00	00 00 D0 02	00 00 00 00	00 00 F8 02D.....ø.
0140h:	00 00 EC 02	00 00 00 00	00 00 80 00	45 78 69 74	..i.....E.Exit
0150h:	50 72 6F 63	65 73 73 00	6B 65 72 6E	65 6C 33 32	Process.kernel32
0160h:	2E 64 6C 6C	00 00 62 02	77 73 70 72	69 6E 74 66	.dll..b.wsprintf
0170h:	41 00 9D 01	4D 65 73 73	61 67 65 42	6F 78 41 00	A...MessageBoxA.
0180h:	75 73 65 72	33 32 2E 64	6C 6C 00 00	00 00 00 00	user32.dll....

(5) 修改函数调用内容。

在之前的步骤中只是删掉了只读数据节的节表内容，在此要对只读数据节进行删除，首先只读数据节中保存有字符串“MessageBoxA”及其对应的 dll “user32.dll”。

“user32.dll”占有 12 个字节，在当前 PE 文件中寻找一个连续的不重要的位置存放该数据，经观察比对后发现 0CH-17H 位置保存的数据为文件的时间戳 TimeDateStamp, COFF 文件符号表在文件中的偏移 PointerToSymbolTable 和符号表的数量 NumberOfSymbols 三个变量的大小恰好为 12 个字节，于是用“user32.dll”替换三个变量的内容。

“MessageBoxA”的长度是 14 个字节，在当前文件中查找 14 个字节的不重要的块。在可选文件头中找到五个连续的变量 MajorLinkerVersion 链接器的主版本号， MinorLinkerVersion 链接器的次版本号， SizeOfCode 代码节大小， SizeOfInitializedData 已初始化数大小和 SizeOfUninitializedData 未初始化数大小，这五个变量对于程序的运行并不重要，所以将其替换为“MessageBoxA”将两个字符串替换后的结果如下图所示。



0000h:	4D 5A 11 11	50 45 00 00	4C 01 01 00	75 73 65 72
0010h:	33 32 2E 64	6C 6C 00 00	E0 00 0F 01	0B 01 4D 65
0020h:	73 73 61 67	65 42 6F 78	41 00 00 00	3A 02 00 00
0030h:	F8 01 00 00	6C 02 00 00	00 00 40 00	04 00 00 00

同时由于 ExitProcess 在函数运行过程中并不是必须的函数调用，所以在当前 PE 文件中将其调用的部分删除。删除后的整个 PE 文件如下图所示。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE..L...user
0010h:	33	32	2E	64	6C	6C	00	00	E0	00	0F	01	0B	01	4D	65	32.dll..à.....Me
0020h:	73	73	61	67	65	42	6F	78	41	00	00	00	3A	02	00	00	essageBoxA.....
0030h:	F8	01	00	00	6C	02	00	00	00	00	40	00	04	00	00	00	ø...l.....@.....
0040h:	04	00	00	00	04	00	00	00	00	00	00	00	04	00	00	00
0050h:	00	00	00	00	14	03	00	00	F8	01	00	00	00	00	00	00ø.....
0060h:	02	00	00	00	00	00	10	00	00	10	00	00	00	00	10	00
0070h:	00	10	00	00	00	00	00	00	02	00	00	00	00	00	00	00
0080h:	00	00	00	00	80	02	00	00	3C	00	00	00	2E	74	65	78€...<....tex
0090h:	74	00	00	00	72	00	00	00	F8	01	00	00	74	00	00	00	t...r...ø...t...
00A0h:	F8	01	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69	ø...MyminiEXE,si
00B0h:	7A	65	3A	37	38	38	42	00	CE	E4	B4	F3	D0	C5	B0	B2	ze:788B.îä'ôÐÄ°²
00C0h:	50	45	D7	F7	D2	B5	28	D0	D5	C3	FB	3A	C4	B3	C4	B3	PE×÷Òµ(ÐÕÄû:Ä°Ä³
00D0h:	C4	B3	2C	D1	A7	BA	C5	3A	32	30	30	33	33	32	31	31	Ä³,ÑS°Ä:20033211
00E0h:	30	2A	2A	2A	29	00	D0	02	00	00	00	00	00	00	F8	02	0***)...D.....ø.
00F0h:	00	00	EC	02	00	00	00	00	00	00	BC	02	00	00	00	00	..i.....¼.....
0100h:	00	00	00	00	00	00	DE	02	00	00	6C	02	00	00	C4	02P...l...Ä.
0110h:	00	00	00	00	00	00	00	00	00	00	06	03	00	00	74	02t.
0120h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0130h:	00	00	00	00	00	00	D0	02	00	00	00	00	00	00	F8	02D.....ø.
0140h:	00	00	EC	02	00	00	00	00	00	00							..i.....

（6）只读数据段重构。

首先我们将原本的只读数据段进行删除。因为文件中的对其粒度为 4，所以需要在文件的末尾填充两个字符保证下一部分的起始地址是 4 的倍数保证对齐。

对于 IAT 表我们需要让其指向我们从 user32.dll 中调用的函数“MessageBoxA”因此我们将其设置为 1C 00 00 00 00 00 00 00h。重构后的 PE 文件如下图所示。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE..L...user
0010h:	33	32	2E	64	6C	6C	00	00	E0	00	0F	01	0B	01	4D	65	32.dll..à.....Me
0020h:	73	73	61	67	65	42	6F	78	41	00	00	00	3A	02	00	00	essageBoxA.....
0030h:	F8	01	00	00	6C	02	00	00	00	00	40	00	04	00	00	00	ø...l.....@.....
0040h:	04	00	00	00	04	00	00	00	00	00	00	00	04	00	00	00
0050h:	00	00	00	00	14	03	00	00	F8	01	00	00	00	00	00	00ø.....
0060h:	02	00	00	00	00	00	10	00	00	10	00	00	00	00	10	00
0070h:	00	10	00	00	00	00	00	00	02	00	00	00	00	00	00	00
0080h:	00	00	00	00	80	02	00	00	3C	00	00	00	2E	74	65	78€...<....tex
0090h:	74	00	00	00	72	00	00	00	F8	01	00	00	74	00	00	00	t...r...ø...t...
00A0h:	F8	01	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69	ø...MyminiEXE,si
00B0h:	7A	65	3A	37	38	38	42	00	CE	E4	B4	F3	D0	C5	B0	B2	ze:788B.îä'ôÐÄ°²
00C0h:	50	45	D7	F7	D2	B5	28	D0	D5	C3	FB	3A	C4	B3	C4	B3	PE×÷Òµ(ÐÕÄû:Ä°Ä³
00D0h:	C4	B3	2C	D1	A7	BA	C5	3A	32	30	30	33	33	32	31	31	Ä³,ÑS°Ä:20033211
00E0h:	30	2A	2A	2A	29	00	00	00	1C	00	00	00	00	00	00	00	0***)...D.....ø.
00F0h:																	..i.....

对于 IDT 表，我们需要让其 OriginalFirstThunk 变量和 FirstThunk 变量指向 IAT 的位置，即为 E8 00 00 00h。将变量 ForwarderChain 变量指向第一个 API 传递器链表）即为 0C 00 00 00H。最终将 PE 文件修改结果如下图所示。

(7) 补充函数代码。

由于我们在第四步中将程序的代码删除，所以在此需要重新写程序的代码部分。首先我们先把当前 PE 文件中能够放置数据的不重要的位置标出来如下图所示。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE..L...user
0010h:	33	32	2E	64	6C	6C	00	00	E0	00	0F	01	0B	01	4D	65	32.dll...à.....Me
0020h:	73	73	61	67	65	42	6F	78	41	00	00	00	3A	02	00	00	ssageBoxA...:...
0030h:	11	11	11	11	11	11	11	11	00	00	40	00	04	00	00	00@.....
0040h:	04	00	00	00	11	11	11	11	11	11	11	11	04	00	00	00ø.....
0050h:	11	11	11	11	11	11	11	11	F8	01	00	00	11	11	11	11ø.....
0060h:	02	00	00	00	11	11	11	11	11	11	11	11	11	11	11	11ø.....
0070h:	11	11	11	11	11	11	11	11	02	00	00	00	11	11	11	11ø.....
0080h:	11	11	11	11	80	02	00	00	3C	00	00	00	2E	74	65	78€...<...tex
0090h:	74	00	00	00	11	11	11	11	F8	01	00	00	11	11	11	11	t.....ø.....
00A0h:	F8	01	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69	ø...MyminiEXE,si
00B0h:	7A	65	3A	37	38	38	42	00	CE	E4	B4	F3	D0	C5	B0	B2	ze:788B.îä'ôðÄ°²
00C0h:	50	45	D7	F7	D2	B5	28	D0	D5	C3	FB	3A	C4	B3	C4	B3	PE×÷Ôp(ðÖÄû:Ä³Ä³
00D0h:	C4	B3	2C	D1	A7	BA	C5	3A	32	30	30	33	33	32	31	31	Ä³,Ñs°Ä:20033211
00E0h:	30	2A	2A	2A	29	00	00	00	1C	00	00	00	00	00	00	00	0***).
00F0h:	E8	00	00	00	00	00	00	00	00	00	00	00	0C	00	00	00	è.....
0100h:	E8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	è.....

将原 PE 文件使用 OllyICE 打开后如下图所示。

00401042	68 40100000	push	1040	
00401047	68 00104000	push	00401000	ASCII "MyminiExe,Size:***B"
0040104C	68 14104000	push	00401014	
00401051	6A 00	push	0	
00401053	E8 08000000	call	<jmp.&user32.MessageBoxA>	
00401058	6A 00	push	0	
0040105A	E8 07000000	call	<jmp.&kernel32.ExitProcess>	
0040105F	CC	int3		
00401060	- FF25 08204000	jmp	dword ptr [&user32.MessageBoxA]	user32.MessageBoxA
00401066	- FF25 00204000	jmp	dword ptr [&kernel32.ExitProcess]	KERNEL32.ExitProcess
0040106C	0000	add	byte ptr [eax], al	
0040106E	0000	add	byte ptr [eax], al	

仿照原 PE 文件的汇编指令编写一段汇编指令如下图所示。

1	push	0x1040	
2	push	0x4000A4	//弹窗标题地址
3	push	0x4000B8	//弹窗内容地址
4	push	0x0	
5	call	x	
6	x: jmp	dword ptr ds:[0x4000E8]	//IAT地址

将汇编指令逐个翻译成机器码后写入到当前 PE 文件中，并根据其在 PE 文件中的位置调整 jmp 和 call 函数的参数。翻译成的机器码如下图所示

68	40	10	00	00
68	A4	00	40	00
68	B8	00	40	00
6A	00			
E8	XX	XX	XX	XX
FF	25	E8	00	40

在将机器码填入 PE 文件的过程中，由于代码位置不连续所以应该在代码断开的位置补上 `jmp` 指令进行跳转。将机器码填入后的 PE 文件如下图所示。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE...user
0010	33	32	2E	64	6C	6C	00	00	E0	00	0F	01	0B	01	4D	65	32.dll...à....Me
0020	73	73	61	67	65	42	6F	78	41	00	00	00	3A	02	00	00	ssageBoxA...:...
0030	68	40	10	00	00	EB	0D	11	00	00	40	00	04	00	00	00	h@...ë....@.....
0040	04	00	00	00	68	A4	00	40	00	EB	1A	11	04	00	00	00h@.@.ë.....
0050	11	11	11	11	11	11	11	11	F8	01	00	00	11	11	11	11ø.....
0060	02	00	00	00	11	68	B8	00	40	00	6A	00	E8	0B	00	00h,.@.j.è...
0070	00	EB	09	00	11	11	11	11	02	00	00	00	FF	25	E8	00	.ë.....ÿ%è.
0080	40	00	00	00	F0	00	00	00	28	00	00	00	2E	74	65	78	@...ð...{...tex
0090	74	00	00	00	11	11	11	11	80	02	00	00	11	11	11	11	t.....€.....
00A0	3C	00	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69	<...MyminiEXE,si
00B0	7A	65	3A	32	35	36	42	00	CE	E4	B4	F3	D0	C5	B0	B2	ze:256B.îä'óðÄ°²
00C0	50	45	D7	F7	D2	B5	28	D0	D5	C3	FB	3A	C4	B3	C4	B3	PE×÷ðµ(ðÖÄû:Ä³Ä³
00D0	C4	B3	2C	D1	A7	BA	C5	3A	32	30	30	33	33	32	31	31	Ä³,Ñš°Ä:20033211
00E0	30	2A	2A	2A	29	00	00	00	1C	00	00	00	00	00	00	00	0***).....
00F0	E8	00	00	00	00	00	00	00	00	00	00	00	0C	00	00	00	è.....
0100	E8	00	00	00	00	00	00	00	00	00	00	00					è.....

其中 `call` 指令的操作码为 `7CH-71H=0BH`

(8) 修改参数

首先修改变量 `SizeOfOptionalHeader` 整个可选 NT 头的大小，在第 2 步中删除了 14 个目录项即 $14 \times 8 = 112$ 个字节，所以该变量的值应该改为 `E0-70=70H`。修改后的结果如下图所示。

0000h:	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE..l...user
0010h:	33	32	2E	64	6C	6C	00	00	70	00	0F	01	0B	01	4D	65	32.dll...p.....Me

然后修改变量 `AddressOfEntryPoint` 的值，将其指向程序代码的起始位置。

0020h:	73	73	61	67	65	42	6F	78	41	00	00	00	30	00	00	00	ssageBoxA...0...
0030h:	68	40	10	00	00	EB	0D	11	00	00	40	00	04	00	00	00	h@...ë....@.....

然后修改变量 `SizeOfHeaders` 指向第一个节开始的位置，在这里让其指向程序开始地址。

0050h:	11	11	11	11	11	11	11	11	30	00	00	00	11	11	11	11
--------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

然后是修改导入表的 `RVA` 和 `Size`，`RVA` 指向 `F0` 位置，大小为 28 个字节。

0080h:	40	00	11	11	F0	00	00	00	28	00	00	00	2E	74	65	78
--------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

最后修改节表的 `VirtualAddress` 和 `PointerToRawData` 两个变量的值使其指向 `A4H` 位置。

0090	74	00	00	00	11	11	11	11	A4	00	00	00	11	11	11	11
00A0	A4	00	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69

4.结果展示

最终修改后的 PE 文件如下图所示

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000	4D	5A	11	11	50	45	00	00	4C	01	01	00	75	73	65	72	MZ..PE..L...user
0010	33	32	2E	64	6C	6C	00	00	70	00	0F	01	0B	01	4D	65	32.dll..p....Me
0020	73	73	61	67	65	42	6F	78	41	00	00	00	30	00	00	00	ssageBoxA...0...
0030	68	40	10	00	00	EB	0D	11	00	00	40	00	04	00	00	00	h@...ë....@.....
0040	04	00	00	00	68	A4	00	40	00	EB	1A	11	04	00	00	00h@. @.ë.....
0050	11	11	11	11	11	11	11	11	30	00	00	00	11	11	11	110.....
0060	02	00	00	00	11	68	B8	00	40	00	6A	00	E8	0B	00	00h. @.j.è...
0070	00	EB	09	00	11	11	11	11	02	00	00	00	FF	25	E8	00	.ë.....ÿ%è.
0080	40	00	00	00	F0	00	00	00	28	00	00	00	2E	74	65	78	@...@... (....tex
0090	74	00	00	00	11	11	11	11	A4	00	00	00	11	11	11	11	t.....@.....
00A0	A4	00	00	00	4D	79	6D	69	6E	69	45	58	45	2C	73	69	@...MyminiEXE,si
00B0	7A	65	3A	32	36	38	42	00	CE	E4	B4	F3	D0	C5	B0	B2	ze:268B.Îä´óÐÅ°²
00C0	50	45	D7	F7	D2	B5	28	D0	D5	C3	FB	3A	D5	D4	B2	AE	PE×÷ðµ(ÐÐÃû:ÖÖ²@
00D0	D9	B6	2C	D1	A7	BA	C5	3A	32	30	32	31	33	30	32	31	Ù¶,Ñ§°Å:20213021
00E0	38	31	31	35	36	29	00	00	1C	00	00	00	00	00	00	00	81156).....
00F0	E8	00	00	00	00	00	00	00	00	00	00	00	0C	00	00	00	è.....
0100	E8	00	00	00	00	00	00	00	00	00	00	00					è.....

执行该 PE 文件运行结果如下图所示

