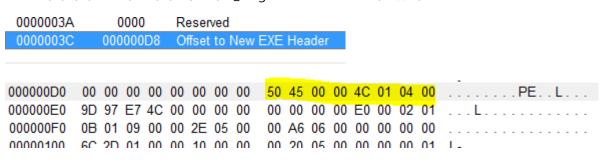


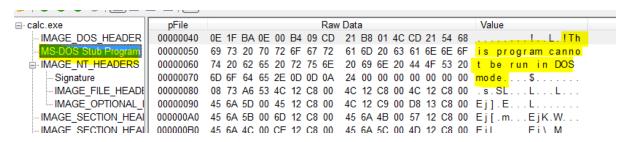
계산기 파일은 window 검색 창에서 cal을 검색해서 파일 경로 보기 후 그 위치로 가서 찾았다.



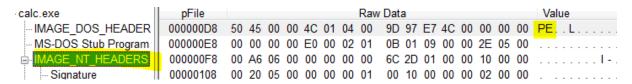
DOS 헤더에서 PE 파일이기 때문에 e_magic 값으로 MZ를 확인했다.



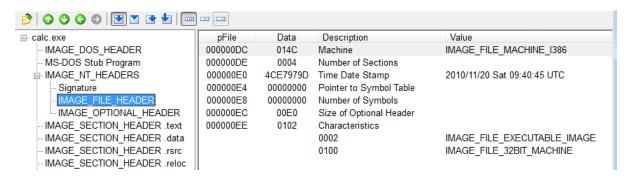
DOS 헤더에서 NT 헤더가 시작되는 부분(e_ifanew)을 확인했다.



DOS_STUB 헤더에서 다음과 같이 윈도우 프로그램을 CMD에서 실행했을 경우 출력해줄 에러 메시지가 저장된 것을 확인했다.



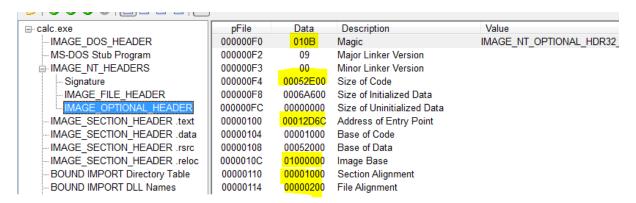
NT 헤더에서 PE 파일의 시그니처인 45 50 값을 확인했다. (자료형 DWORD



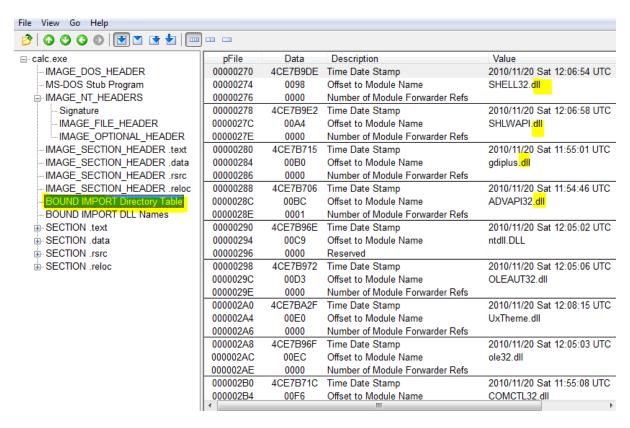
NT 헤더에 FILE 헤더 구조체에 정보를 확인했다.

컴파일한 기계와 시간 정보인 Timestamp와 machine를 확인하고,

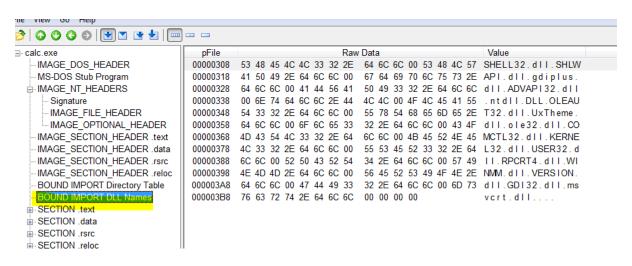
섹션의 개수가 4개이고, characteristics를 통해 0002 – 실행 가능하고 0100 - 32bit machine인 것을 확인했다.



Optional 헤더 구조체에서 magic 값이 010B, 즉 32비트인 것을 알 수 있고 실제 코드가 올라가는 .TEXT 영역의 크기를 SIZE OF CODE를 통해 알 수 있고 프로그램의 RVA를 ADDRESS OF ENTRY POINT를 통해 알 수 있고 실제 디버거에서 사용되는 주소인 VA를 구하기 위해 필요한 IMAGEBASE의 값을 알 수 있고 SECTION ALIGNMENT를 통해 메모리에서의 섹션의 최소 단위를 알 수 있고



Import table을 통해서 exe 파일이 실행되기 위해 필요한 pe 파일인 dll등과 같은 파일의 정보를 확인할 수 있다.



Import dll names에서는 이름과 같이 위에서 본 dll 파일들의 이름들만을 확인할 수 있다.

calc.exe	pFile	Data	Description	Value
IMAGE_DOS_HEADER	000001D0	2E 74 65 78	Name	.text
MS-DOS Stub Program	000001D4	74 00 00 00		
- IMAGE_NT_HEADERS	000001D8	00052CA1	Virtual Size	
- Signature	000001DC	00001000	RVA	
IMAGE_FILE_HEADER	000001E0	00052E00	Size of Raw Data	
IMAGE_OPTIONAL_HEADER	000001E4	00000400	Pointer to Raw Data	
IMAGE_SECTION_HEADER .text	000001E8	00000000	Pointer to Relocations	
IMAGE_SECTION_HEADER .data	000001EC	00000000	Pointer to Line Numbers	
IMAGE_SECTION_HEADER .rsrc	000001F0	0000	Number of Relocations	
IMAGE_SECTION_HEADER .reloc	000001F2	0000	Number of Line Numbers	
BOUND IMPORT Directory Table	000001F4	60000020	Characteristics	
BOUND IMPORT DLL Names			00000020	IMAGE_SCN_CNT_CODE
SECTION .text			20000000	IMAGE_SCN_MEM_EXECUTE
⊕ SECTION .data			40000000	IMAGE_SCN_MEM_READ
SECTION .rsrc				

Section 헤더의 .text 부분에서 다음과 같이 .text 영역이라는 name 값을 확인할 수 있고 .text 섹션의 크기를 virtual size로 확인할 수 있고

Size of raw data값을 통해 파일에서의 섹션 크기를 알 수 있고,

(이 값은 optional 헤더에서 확인한 file alignment의 배수임)

Pointer to raw data를 통해 파일에서 .text의 시작 위치를 알 수 있고

Characteristics를 통해 실행, 읽기가 가능하고 코드 섹션(.text)이라는 것을 확인할 수 있었다.

000001C0	UU																
000001D0	2E	74	65	78	74	00	00	00	A1	2C	05	00	00	10	00	00	. <mark>text</mark> .,
000001E0	00	2E	05	00	00	04	00	00	00	00	00	00	00	00	00	00	
000001F0	00	00	00	00	20	00	00	60	2E	64	61	74	61	00	00	00	` <mark>.data.</mark>
																	.@@B 2
00000210	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	C0	<u></u> @
00000220	2E	72	73	72	63	00	00	00	98	27	06	00	00	90	05	00	<mark>.rsrc.</mark> '
00000230	00	28	06	00	00	74	05	00	00	00	00	00	00	00	00	00	. (t <mark></mark>
00000240	00	00	00	00	40	00	00	40	2E	72	65	6C	6F	63	00	00	
00000250	3C	3B	00	00	00	C0	0B	00	00	3C	00	00	00	9C	0B	00	<;

.text 영역의 Pointer to raw data 값 위치로 가보니 다음과 같이 여러 섹션들의 시작 위치들이 모여있는 것을 확인할 수 있었다.

-IMAGE_SECTION_HEADER .data
-IMAGE_SECTION_HEADER .rsrc
-IMAGE_SECTION_HEADER .reloc
BOUND IMPORT Directory Table
BOUND IMPORT DLL Names
- SECTION .text
- SECTION .data
SECTION rere

00000214 00000218 0000021A 0000021C	00000000 0000 0000 C0000040	Pointer to Line Numbers Number of Relocations Number of Line Numbers Characteristics 00000040 40000000
		40000000
		80000000

IMAGE_SCN_CNT_INITIALIZED_DATA IMAGE_SCN_MEM_READ IMAGE_SCN_MEM_WRITE

.data 섹션의 정보

IMAGE_SECTION_HEADER_rsrc IMAGE_SECTION_HEADER_reloc BOUND IMPORT Directory Table BOUND IMPORT DLL Names SECTION .text SECTION .data	00000240 00000242 00000244	0000 0000 0000 40000040	Number of Relocations Number of Line Numbers Characteristics 00000040 40000000	IMAGE_SCN_CNT_INITIALIZED_DATA IMAGE_SCN_MEM_READ
.rsrc 섹션의 정보이다.				

섹션들은 Characteristics 값을 통해서 구분이 가능하다.