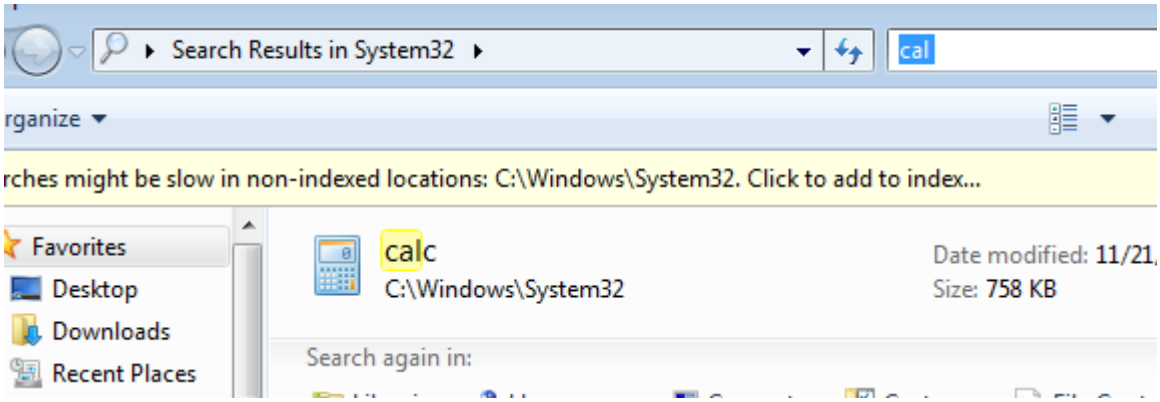
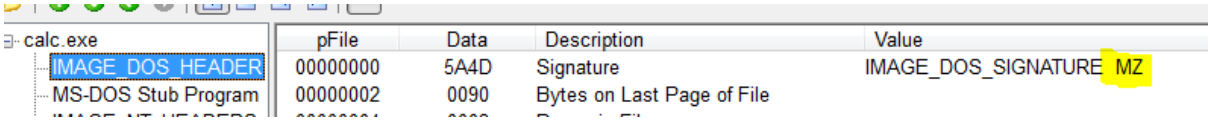
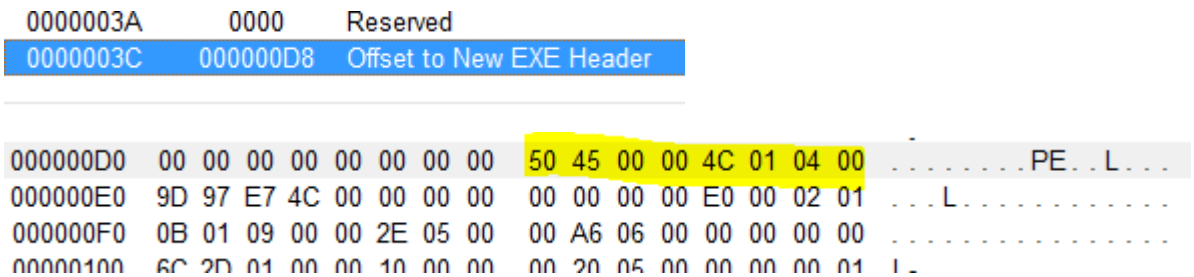


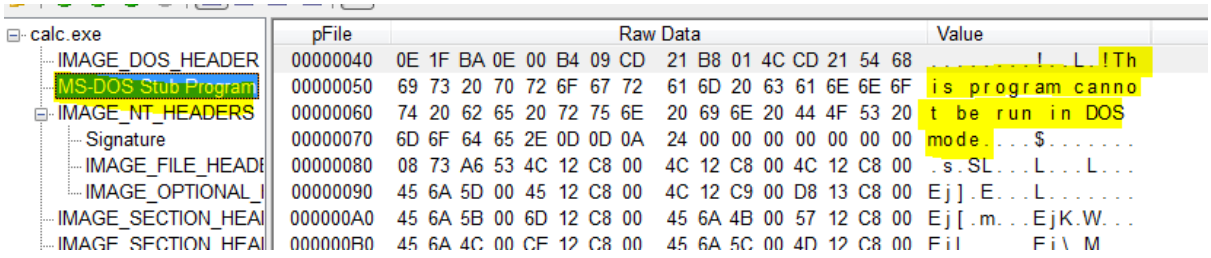

계산기 파일은 window 검색 창에서 cal을 검색해서 파일 경로 보기 후 그 위치로 가서 찾았다.



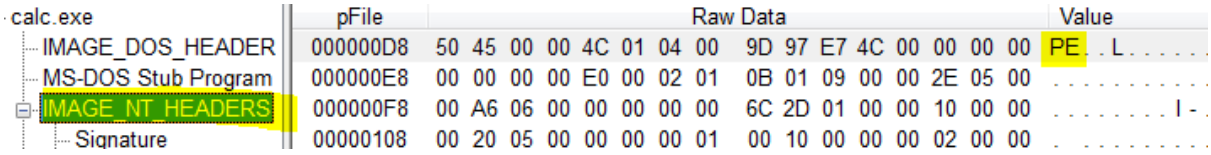

DOS 헤더에서 PE 파일이기 때문에 e_magic 값으로 MZ를 확인했다.



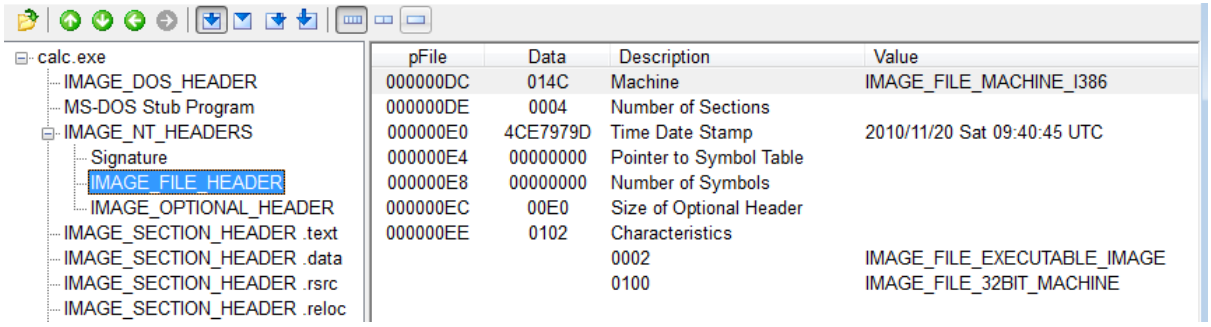

DOS 헤더에서 NT 헤더가 시작되는 부분(e_ifanew)을 확인했다.




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



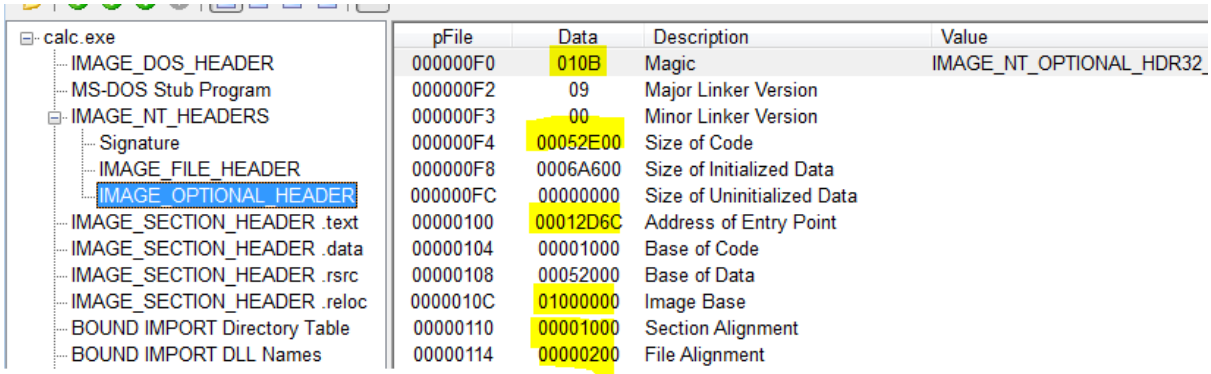

NT 헤더에서 PE 파일의 시그니처인 45 50 값을 확인했다. (자료형 DWORD_




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

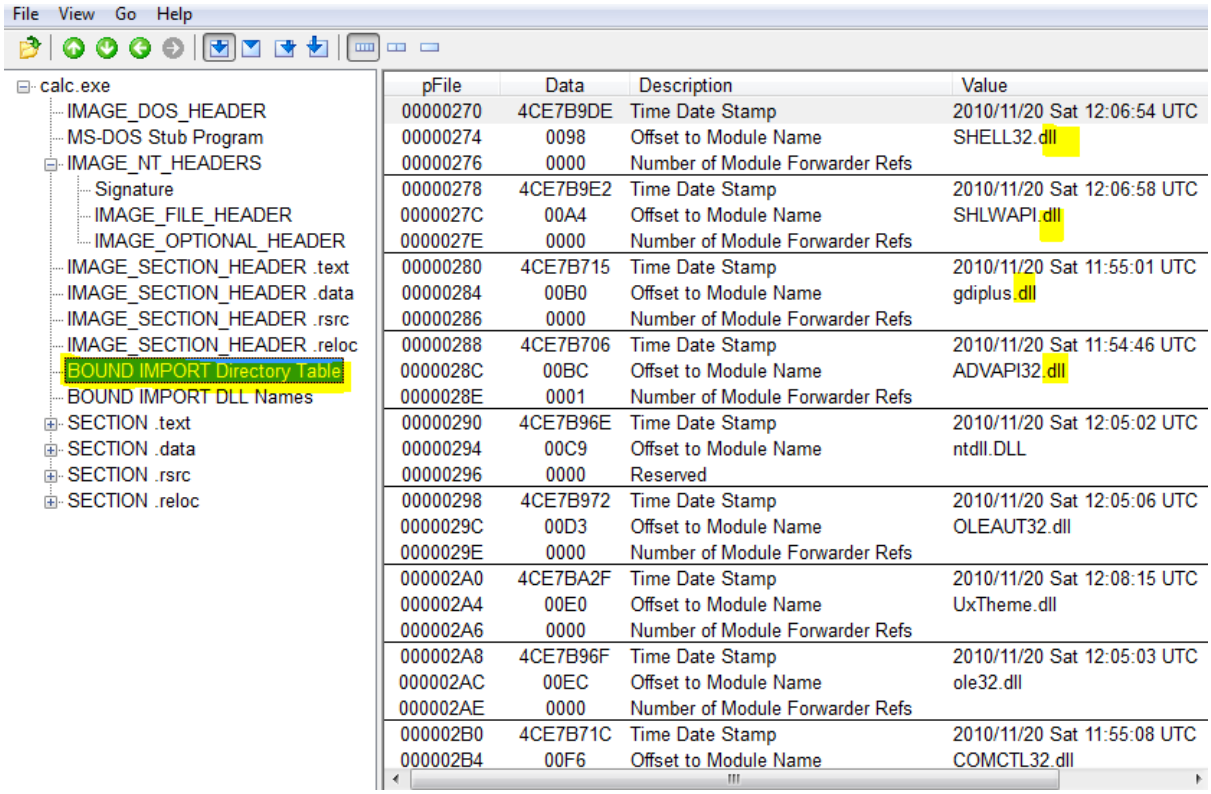
실제 코드가 올라가는 .TEXT 영역의 크기를 SIZE OF CODE를 통해 알 수 있고

프로그램의 RVA를 ADDRESS OF ENTRY POINT를 통해 알 수 있고

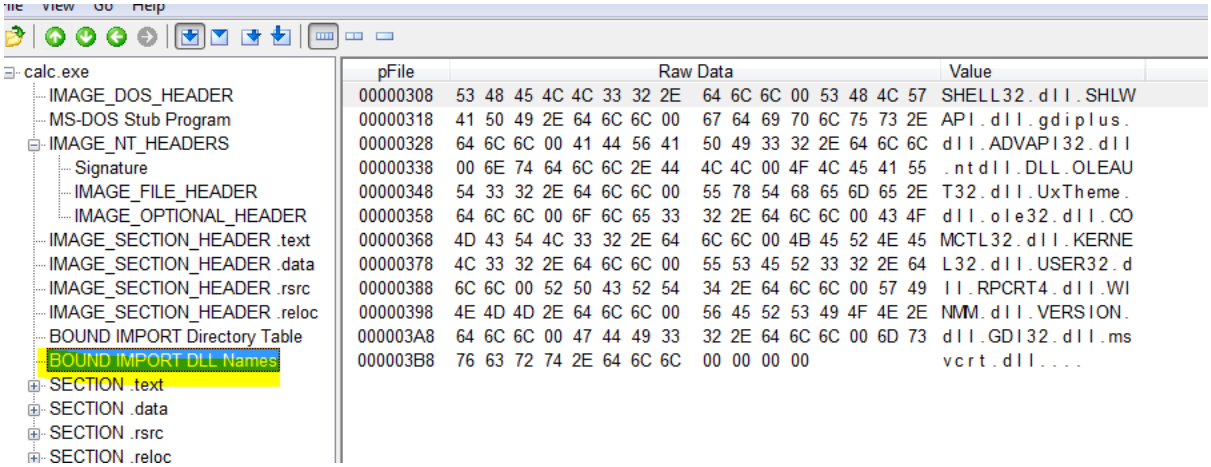
실제 디버거에서 사용되는 주소인 VA를 구하기 위해 필요한 IMAGEBASE의 값을 알 수 있고

SECTION ALIGNMENT를 통해 메모리에서의 섹션의 최소 단위를 알 수 있고

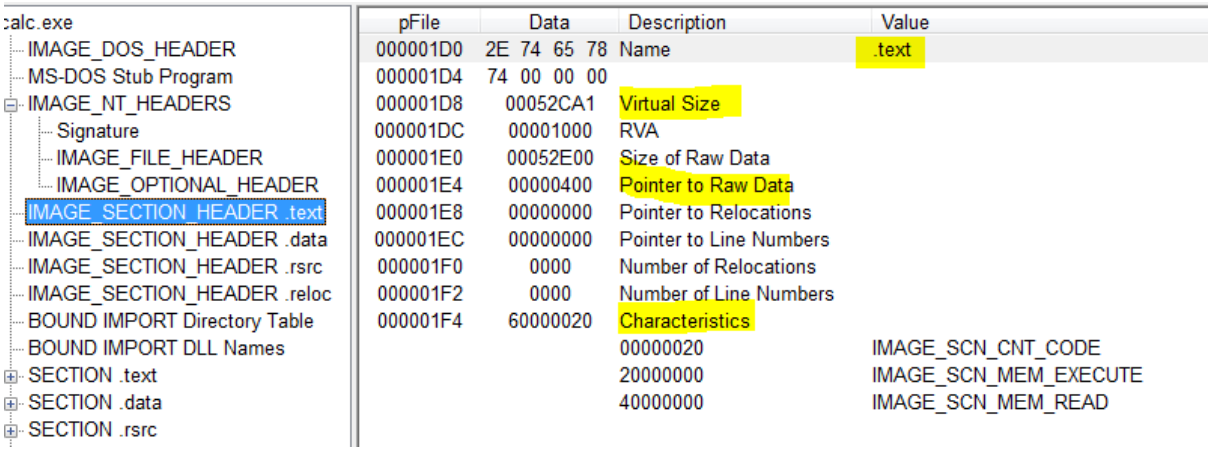
FILE ALIGNMENT를 통해 파일에서의 섹션 최소 단위를 알 수 있다.




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.




Section 헤더의 .text 부분에서 다음과 같이 .text 영역이라는 name 값을 확인할 수 있고

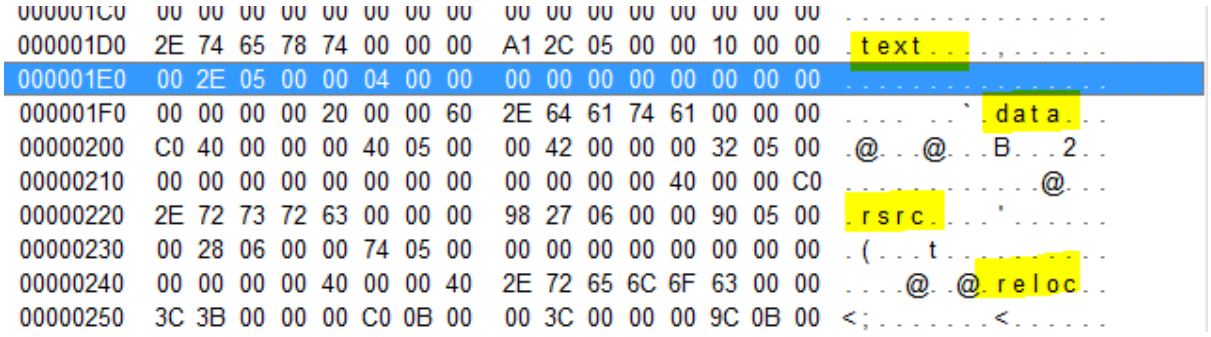
.text 섹션의 크기를 virtual size로 확인할 수 있고

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

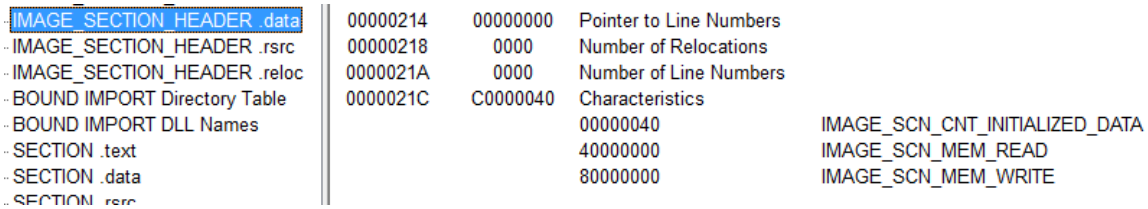
(이 값은 optional 헤더에서 확인한 file alignment의 배수임)

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

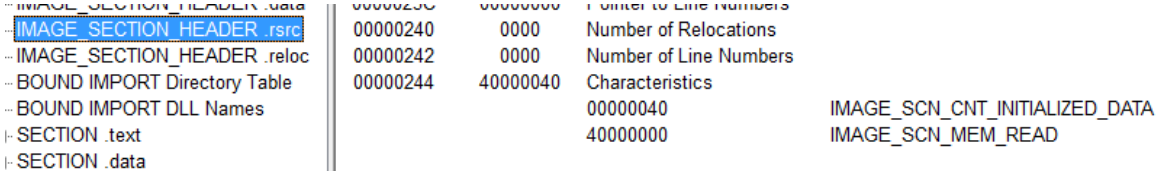
Characteristics를 통해 실행, 읽기가 가능하고 코드 섹션(.text)이라는 것을 확인할 수 있었다.




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




.data 섹션의 정보




.rsrc 섹션의 정보이다.

섹션들은 Characteristics 값을 통해서 구분이 가능하다.