

어셈블리 과제 발표

2021563060 정미주

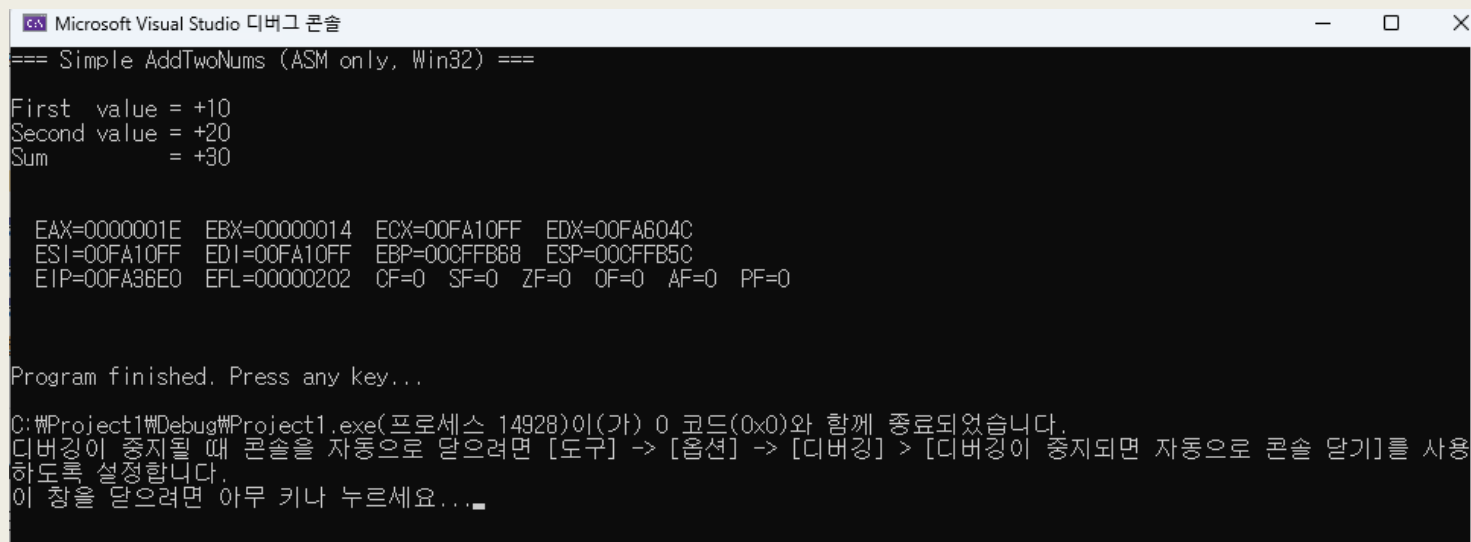
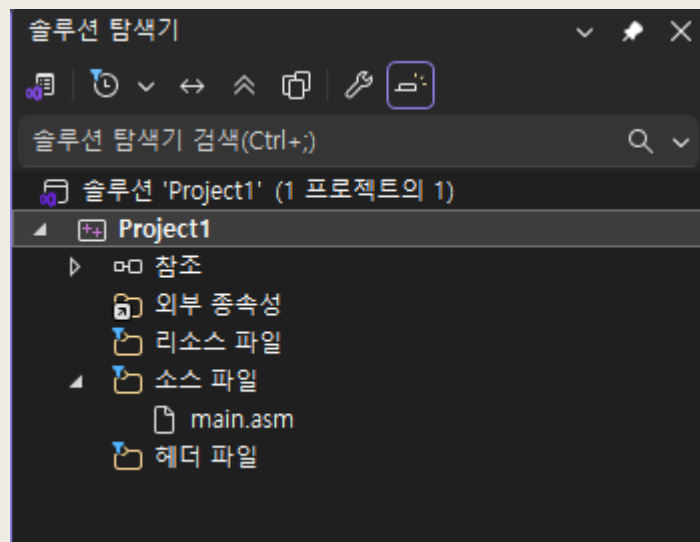
사용 환경

- 운영체제: Windows 10
- 개발도구: Visual Studio 2022
- 언어: C, x86 Assembly (MASM)

프로젝트 진행

- 1. Assembly 단독 실행 프로그램 (.exe)
- 2. 정적 라이브러리 (.lib) 생성
- 3. DLL 프로젝트 생성

1. Assembly 단독 실행 프로그램 (.exe)



두 정수를 더하는 간단한 프로그램을 구현

- main.asm 내부에서 직접 값(push 10, push 20)을 스택에 전달
- AddTwoNums 함수 호출
- 결과 출력

2. 정적 라이브러리 (.lib) 생성

- Assembly 함수(AddTwoNums)를 정적 라이브러리(AddLib.lib) 형태로 분리

--- <구성> ---

AddLib 프로젝트

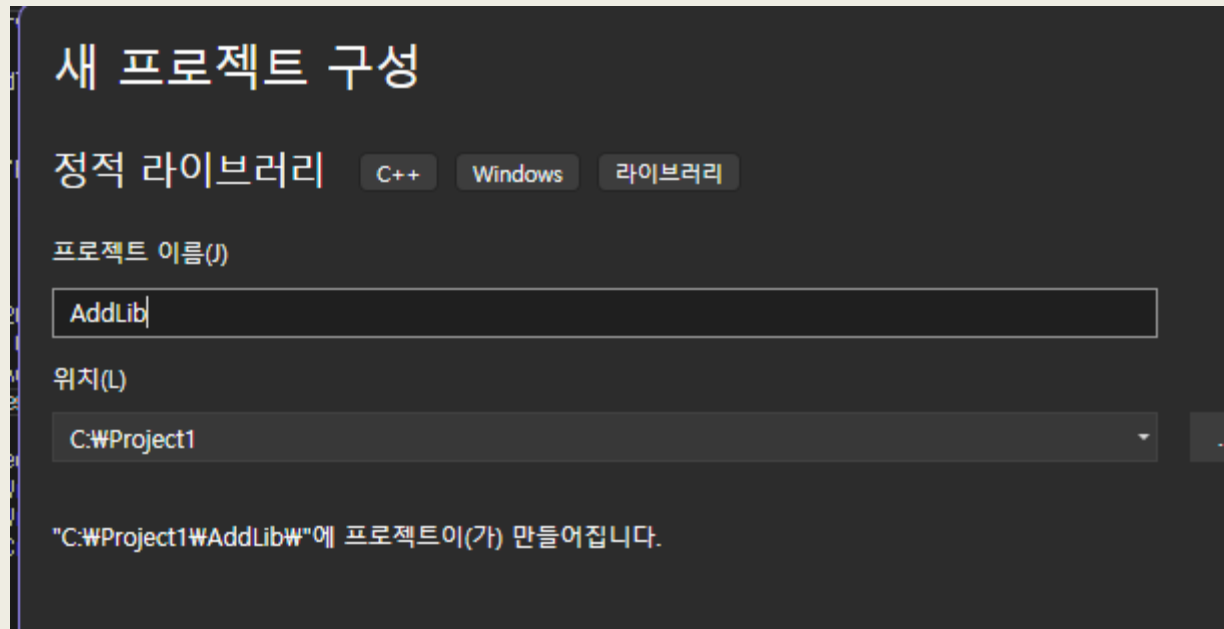
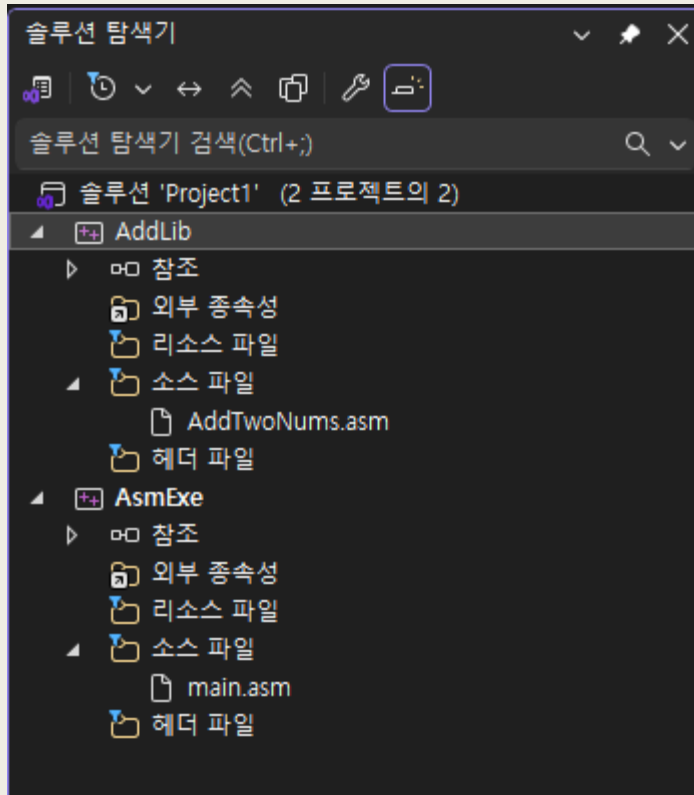
- AddTwoNums.asm
- 결과물: AddLib.lib

AsmExe 프로젝트

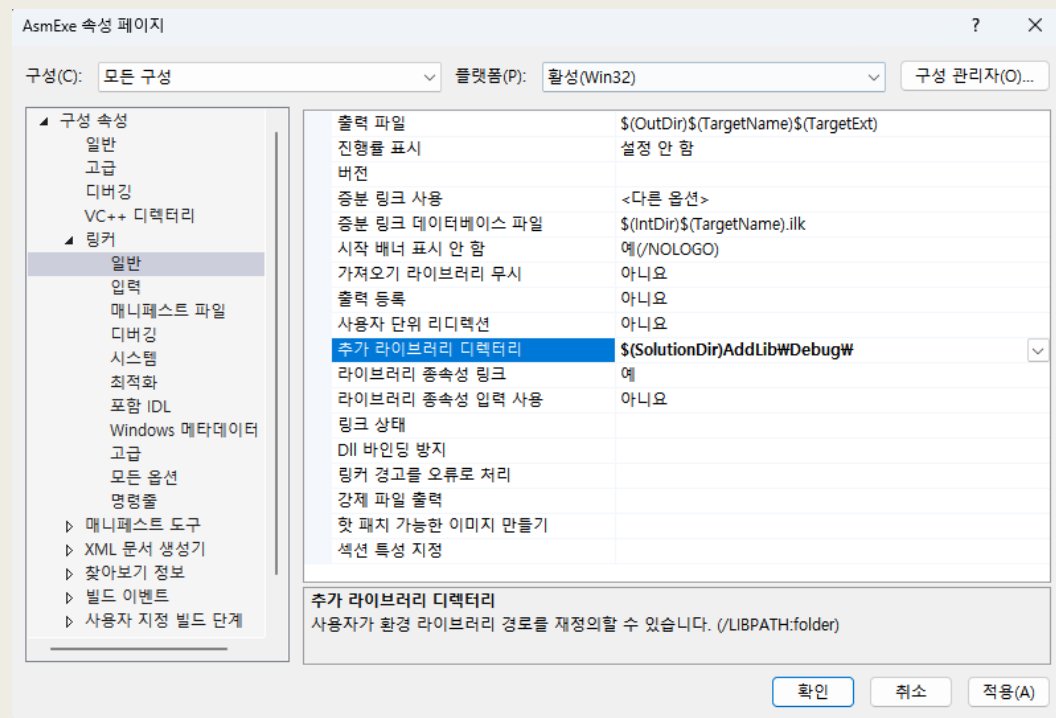
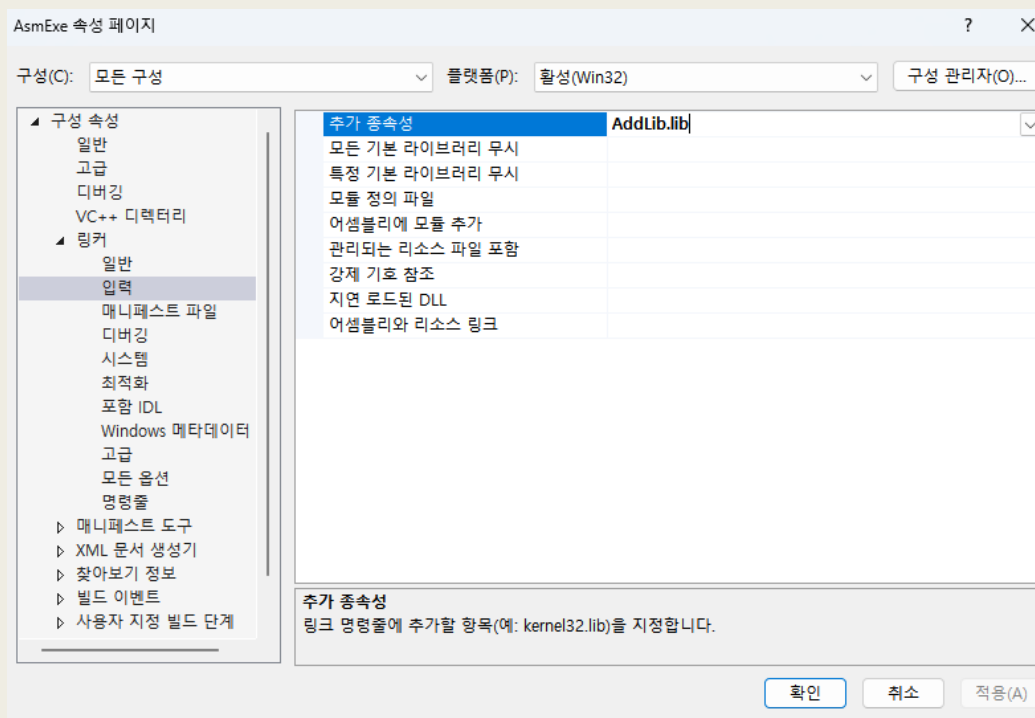
- main.asm

- 링커 설정을 통해 AddLib.lib를 추가 종속성으로 연결

2. 정적 라이브러리 (.lib) 생성



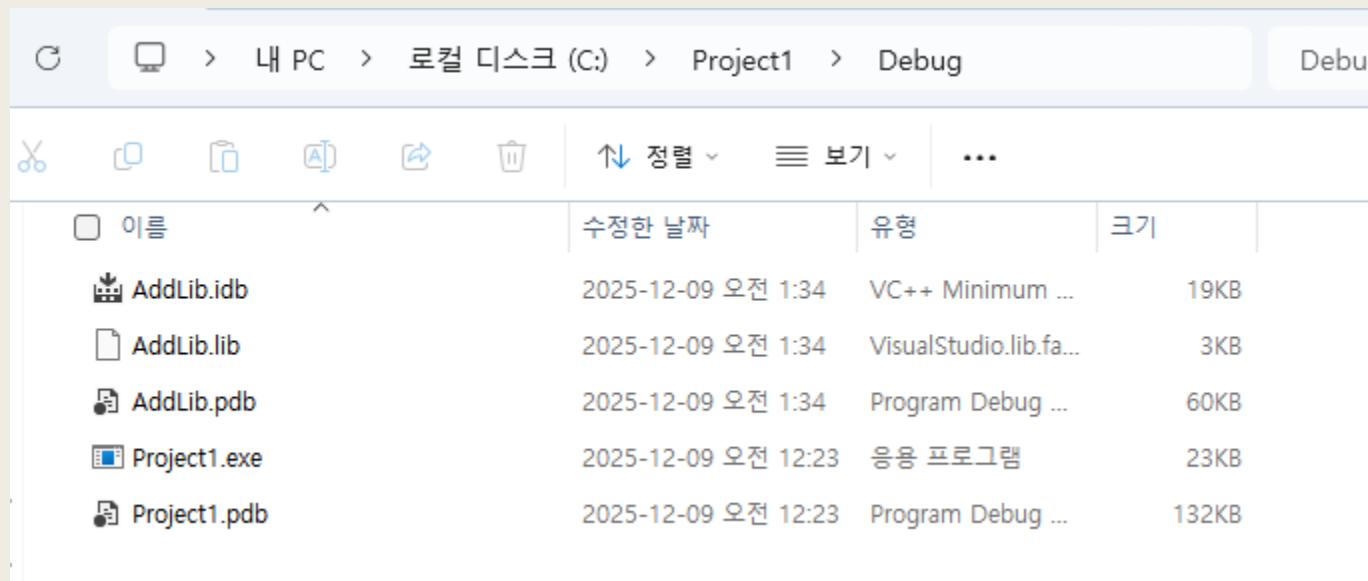
2. 정적 라이브러리 (.lib) 생성



정적 라이브러리를 실행 파일에 연결하는 설정

2. 정적 라이브러리 (.lib) 생성

- Debug 폴더에 AddLib.lib (정적라이브러리) 생성 확인



2-1. 정적 라이브러리 연동 실행

- AsmExe 프로젝트에서 AddLib.lib를 링크하여 Assembly 함수 호출
- 링커 → 입력 → 추가 종속성
- AddLib.lib 정상 빌드 및 실행 확인

2-1. 정적 라이브러리 연동 실행

```
1  INCLUDE Irvine32.inc
2
3  EXTERN AddTwoNums: PROC    ; 정적 라이브러리 함수
4
5  .data
6  msg BYTE "Result from static library = ",0
7
8  .code
9  main PROC
10
11      push 20
12      push 10
13      call AddTwoNums
14      add esp, 8
15
16      mov edx, OFFSET msg
17      call WriteString
18      call WriteInt
19      call CrLf
20
21      exit
22  main ENDP
23  END main
24
```

```
출력 보기 선택(S): 빌드
오전 2:37에 다시 빌드를 시작함...
1>----- 모두 다시 빌드 시작: 프로젝트: AddLib, 구성: Debug Win32 -----
1> Assembling AddTwoNums.asm...
1> dummy.cpp
1> AddLib.vcxproj -> C:\Project1\Debug\AddLib.lib
2>----- 모두 다시 빌드 시작: 프로젝트: AsmExe, 구성: Debug Win32 -----
2> Assembling main.asm...
2> Project1.vcxproj -> C:\Project1\Debug\AsmExe.exe
===== 모두 다시 빌드: 2 성공, 0 실패, 0 건너뛰기 =====
===== 다시 빌드이(가) 오전 2:37에 완료되었으며, 02.584 초이(가) 걸림 =====
```

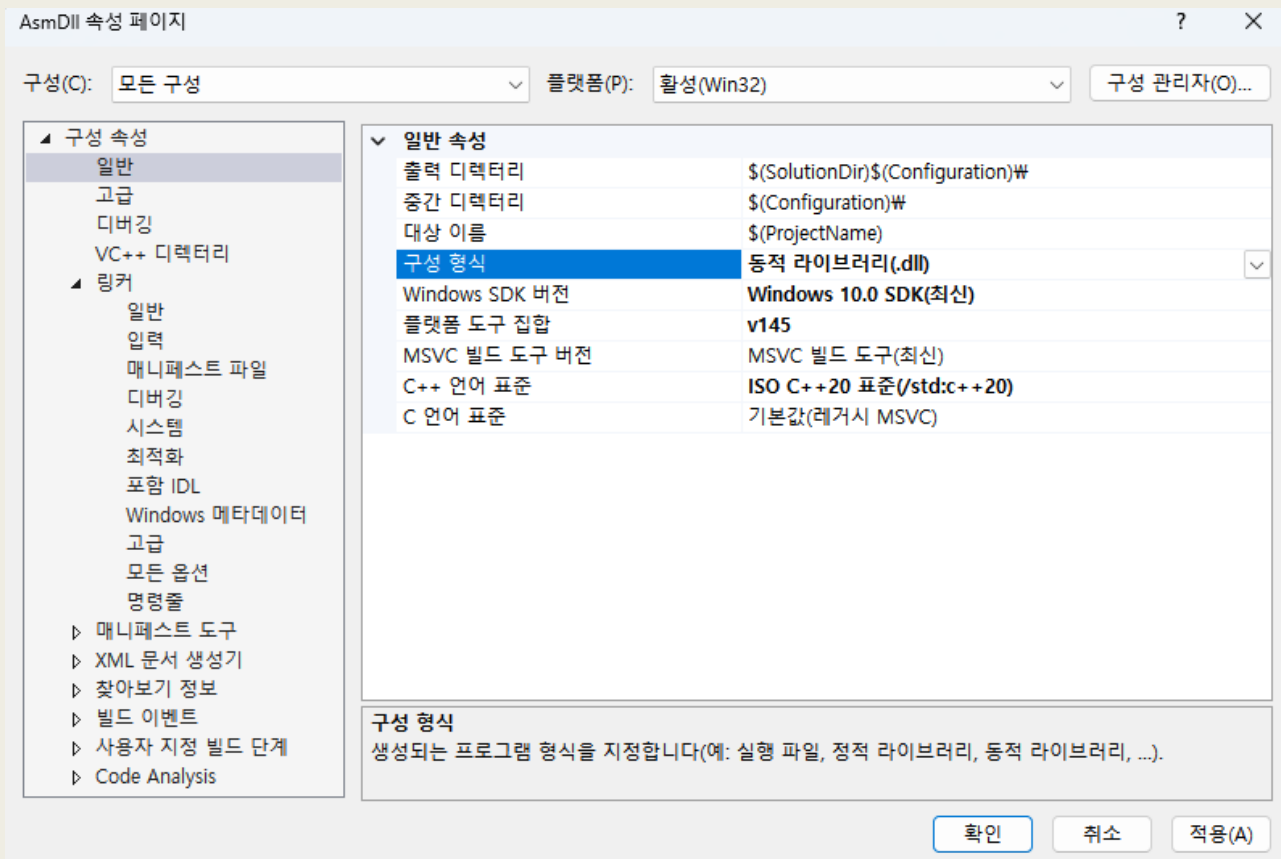
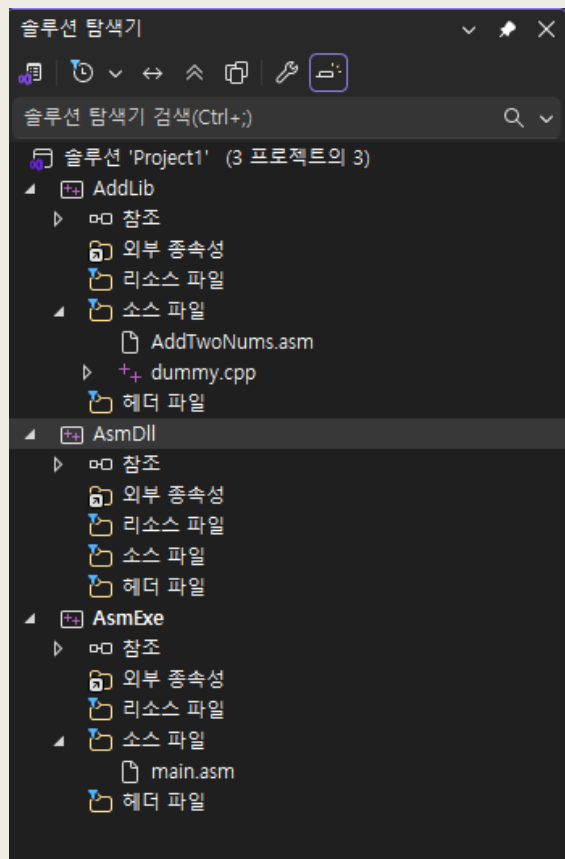
정적 빌드 성공

EXE에서 LIB 함수 호출 코드

3. DLL 프로젝트 생성

- 동일한 Assembly 코드(AddTwoNums.asm)를 사용하여 동적 라이브러리(.dll)로 빌드
- Assembly 코드를 DLL로 빌드하여 외부 C 프로그램에서 호출 가능하도록 구성

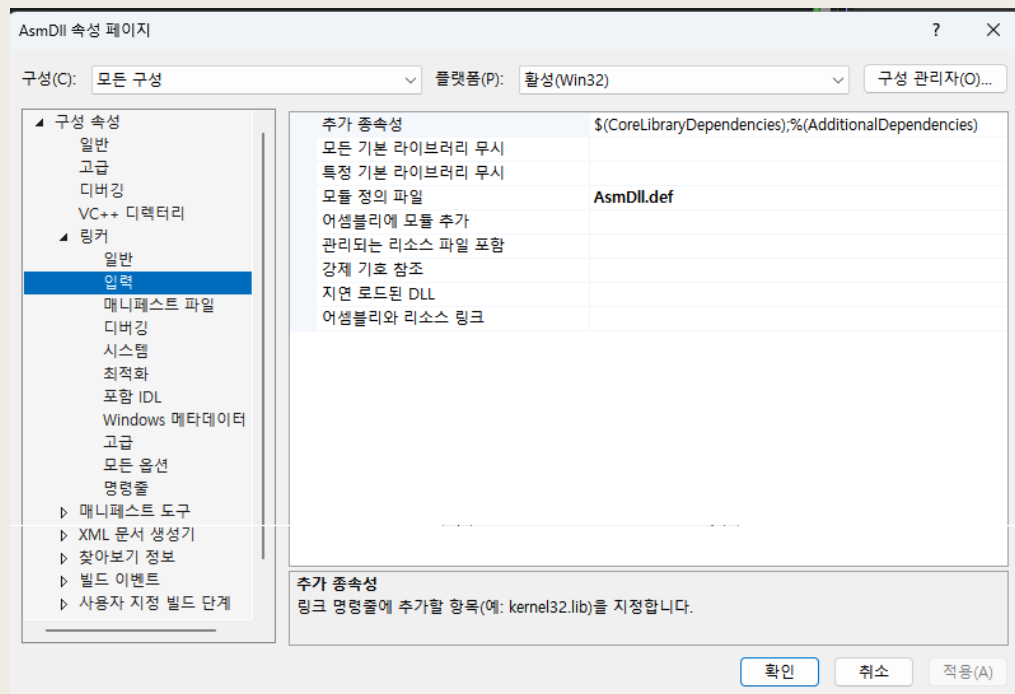
3. DLL 프로젝트 생성



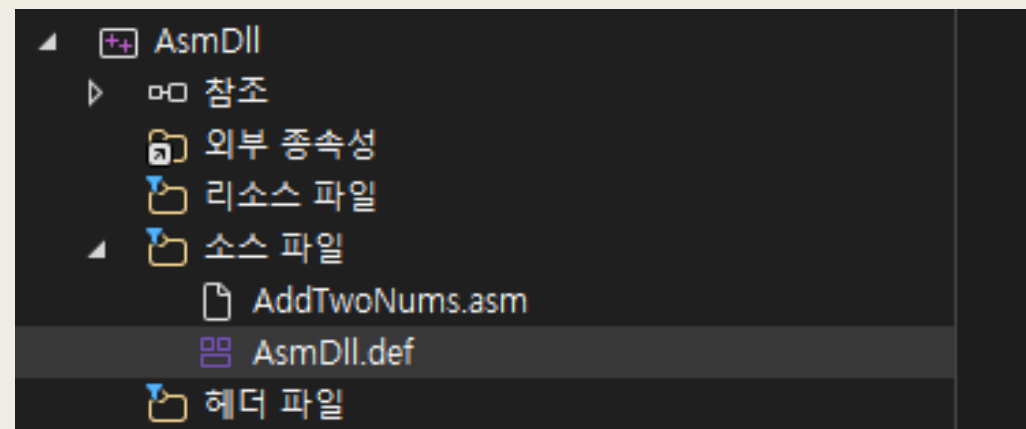
AsmDll 프로젝트 생성 / 속성을 동적 라이브러리(.dll)로 변경

3-1. DLL Export

- DEF 파일 또는 PUBLIC + EXPORT 설정으로 함수 노출
- AddTwoNums.def 사용



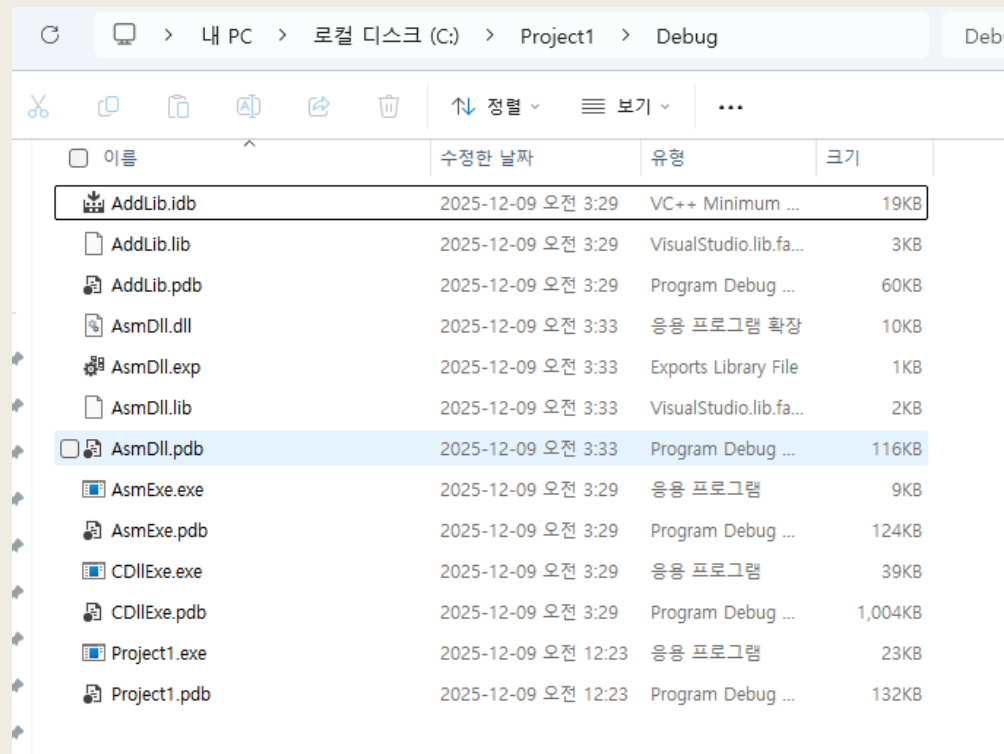
모듈 정의 파일로 AsmDll.def를 지정



AddTwoNums 함수를 EXPORTS 항목에 명시

3-2. DLL 빌드 결과

- 빌드 성공 시 Debug 폴더에 다음 파일 생성
- AsmDll.dll
- AsmDll.lib



이름	수정한 날짜	유형	크기
AddLib.idb	2025-12-09 오전 3:29	VC++ Minimum ...	19KB
AddLib.lib	2025-12-09 오전 3:29	VisualStudio.lib.fa...	3KB
AddLib.pdb	2025-12-09 오전 3:29	Program Debug ...	60KB
AsmDll.dll	2025-12-09 오전 3:33	응용 프로그램 확장	10KB
AsmDll.exp	2025-12-09 오전 3:33	Exports Library File	1KB
AsmDll.lib	2025-12-09 오전 3:33	VisualStudio.lib.fa...	2KB
AsmDll.pdb	2025-12-09 오전 3:33	Program Debug ...	116KB
AsmExe.exe	2025-12-09 오전 3:29	응용 프로그램	9KB
AsmExe.pdb	2025-12-09 오전 3:29	Program Debug ...	124KB
CDllExe.exe	2025-12-09 오전 3:29	응용 프로그램	39KB
CDllExe.pdb	2025-12-09 오전 3:29	Program Debug ...	1,004KB
Project1.exe	2025-12-09 오전 12:23	응용 프로그램	23KB
Project1.pdb	2025-12-09 오전 12:23	Program Debug ...	132KB

출력

출력 보기 선택(S): 빌드

```
1>AddTwoNums.asm(29): warning A6004: procedure argument or local not referenced : reason
1>AddTwoNums.asm(29): warning A6004: procedure argument or local not referenced : reserved
1> C:\Project1\Debug\AsmDll.lib 라이브러리 및 C:\Project1\Debug\AsmDll.exp 개체를 생성하고 있습니다.
1> AsmDll.vcxproj -> C:\Project1\Debug\AsmDll.dll
===== 모두 다시 빌드: 1 성공, 0 실패, 0 건너뛰기 =====
===== 다시 빌드이(가) 오전 3:33에 완료되었으며, 01.267 초이(가) 걸림 =====
```

3-3. C 프로그램에서 DLL 사용

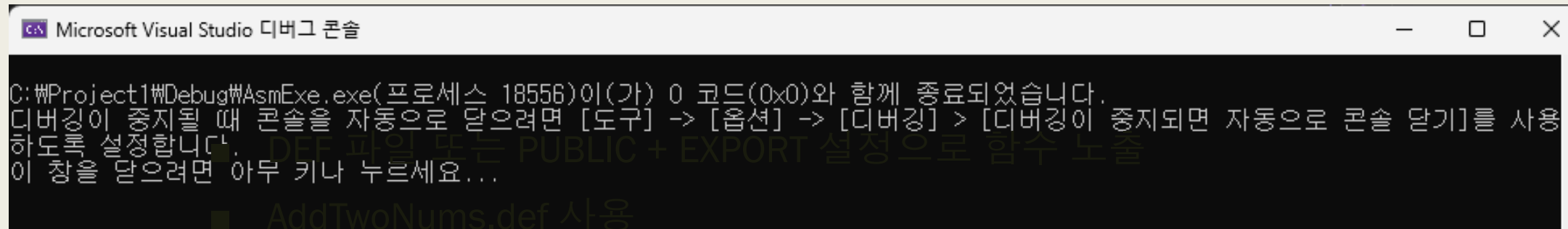
- C 언어 기반 실행 파일에서 Assembly DLL을 호출
- - __stdcall 호출 규약 사용

```
main.c x dummy.cpp AddTwoNums.asm main.asm
CDllExe (전역 범위)
1 #include <stdio.h>
2 #include <windows.h>
3
4 typedef int(__stdcall* AddTwoNumsFunc)(int, int);
5
6 int main(void)
7 {
8     HMODULE hDll = LoadLibraryA("AsmDll.dll");
9     if (!hDll) {
10         printf("DLL load failed\n");
11         return 1;
12     }
13
14     AddTwoNumsFunc AddTwoNums =
15         (AddTwoNumsFunc)GetProcAddress(hDll, "AddTwoNums");
16
17     if (!AddTwoNums) {
18         printf("Function not found\n");
19         return 1;
20     }
21
22     int result = AddTwoNums(10, 20);
23     printf("Result from DLL (ASM) = %d\n", result);
24
25     FreeLibrary(hDll);
26     return 0;
27 }
28
```

main.c

LoadLibrary로 AsmDll.dll을 로드하고
GetProcAddress를 통해 AddTwoNums 함수를 가져옴

최종 실행 결과



```
Microsoft Visual Studio 디버그 콘솔

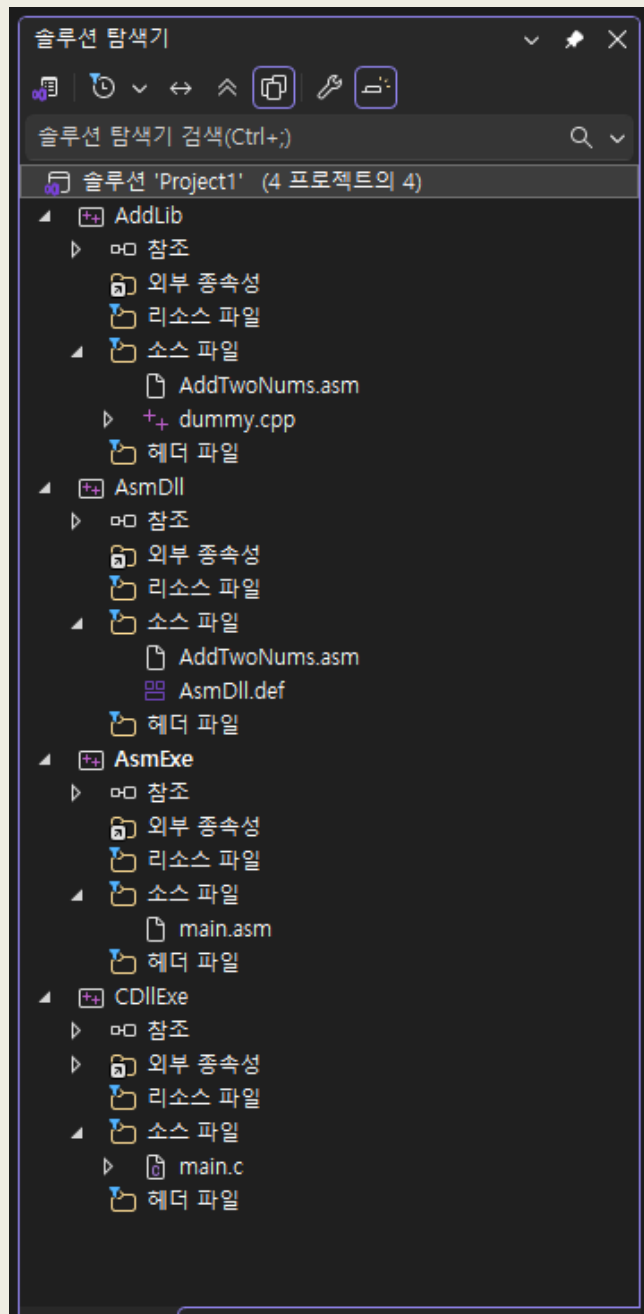
C:\Project1\Debug\AsmExe.exe(프로세스 18556)이(가) 0 코드(0x0)와 함께 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

DEF 파일 또는 PUBLIC + EXPORT 설정으로 함수 노출

■ AddTwoNums.def 사용
```

- AddTwoNums(10, 20)을 호출하면 DLL 내부의 Assembly 코드가 실행되고 결
과가 출력

최종 구조



점수 : 10점

- 핵심을 모두 포함
- 문제없이 완벽 실행

감사합니다.