

# 在 vs2019 中调试汇编控制台程序 (Jin)

## 1. 编写 asm 源程序 (控制台)

利用 vs2019 编写和调试汇编程序需要注意一些事项：

- (1) 为了在vs2019的汇编程序中调用C的函数，需要执行vs2019产生的初始化代码，该初始化代码会调用libcmt.lib中的\_mainCRTStartup()，再由\_mainCRTStartup()调用main()。这就是说，**如果需要在vs2019的汇编程序中调用C函数，必须有入口函数main()（子程序）**，同时将libcmt.lib包含进去（`includelib libcmt.lib`）；
- (2) 由于程序入口是main()，因此不能在程序末尾的end语句上加标号，即不能写 `end start` 之类的语句；
- (3) C函数（如\_mainCRTStartup()、printf()）都是按c规范调用函数，所以必须有语句 `.model flat, c`（不能是stdcall）
- (4) 基本的 C 函数（如 scanf、printf 等）是在 `legacy_stdio_definitions.lib` 中定义的，所以需要在程序中加入语句：`includelib legacy_stdio_definitions.lib`。

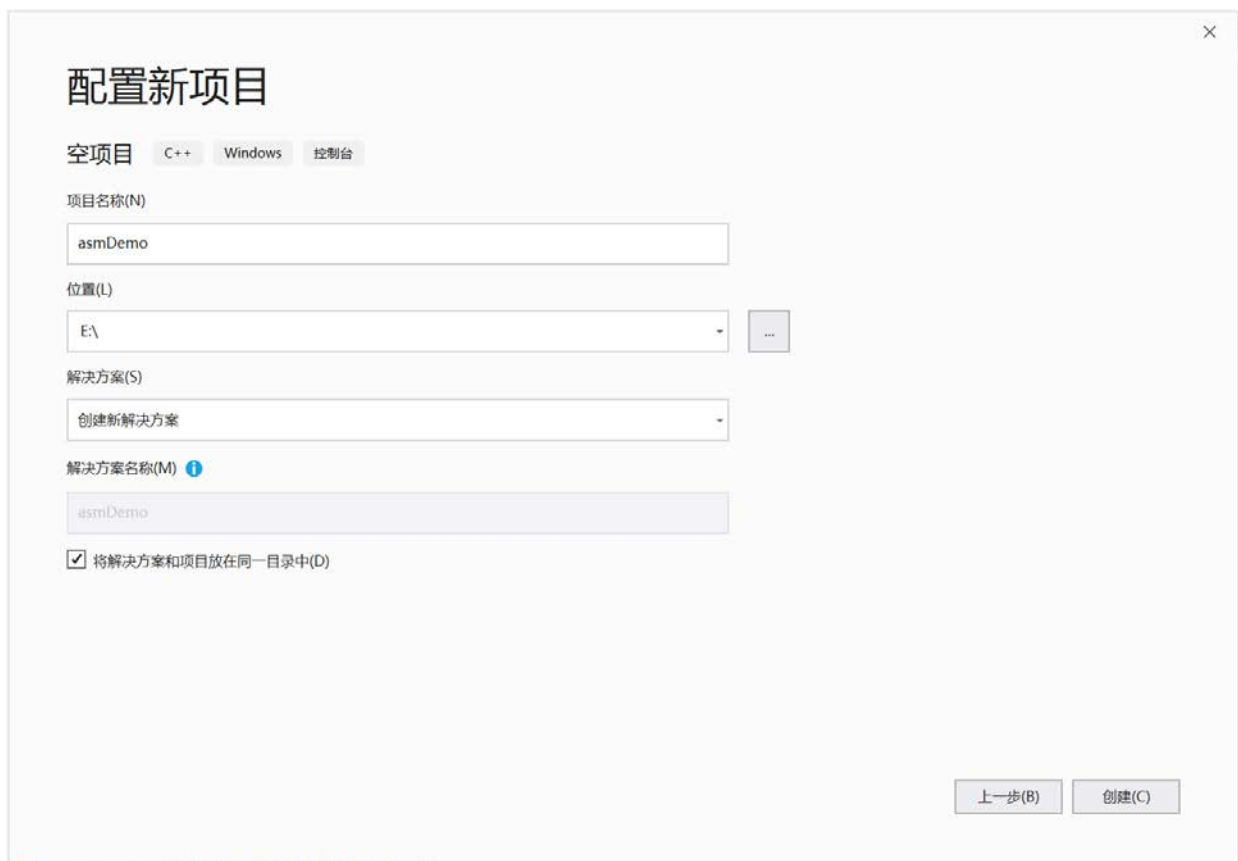
根据上面这些条款，**vs2019\_asmDemo.asm** 演示了在汇编中调用 C 的 `scanf()`和 `printf()`。

**注意：**如果汇编程序不调用任何 C 函数，则可以不定义 `main` 子程序，但需要用标号标明第一条语句，并在 `end` 语句中加上该标号，同时使用 `.model flat, stdcall`。

## 2. 在 vs2019 中调试汇编程序

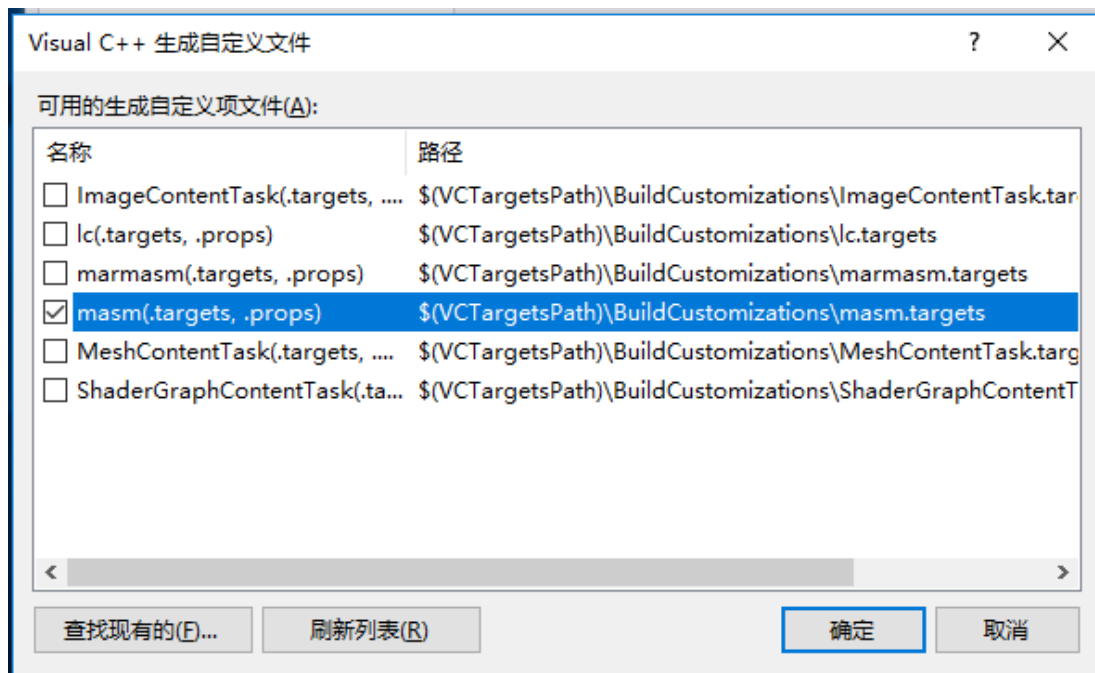
- (1) 运行 vs2019 创建一个空项目（项目名称为 `asmDemo`）；





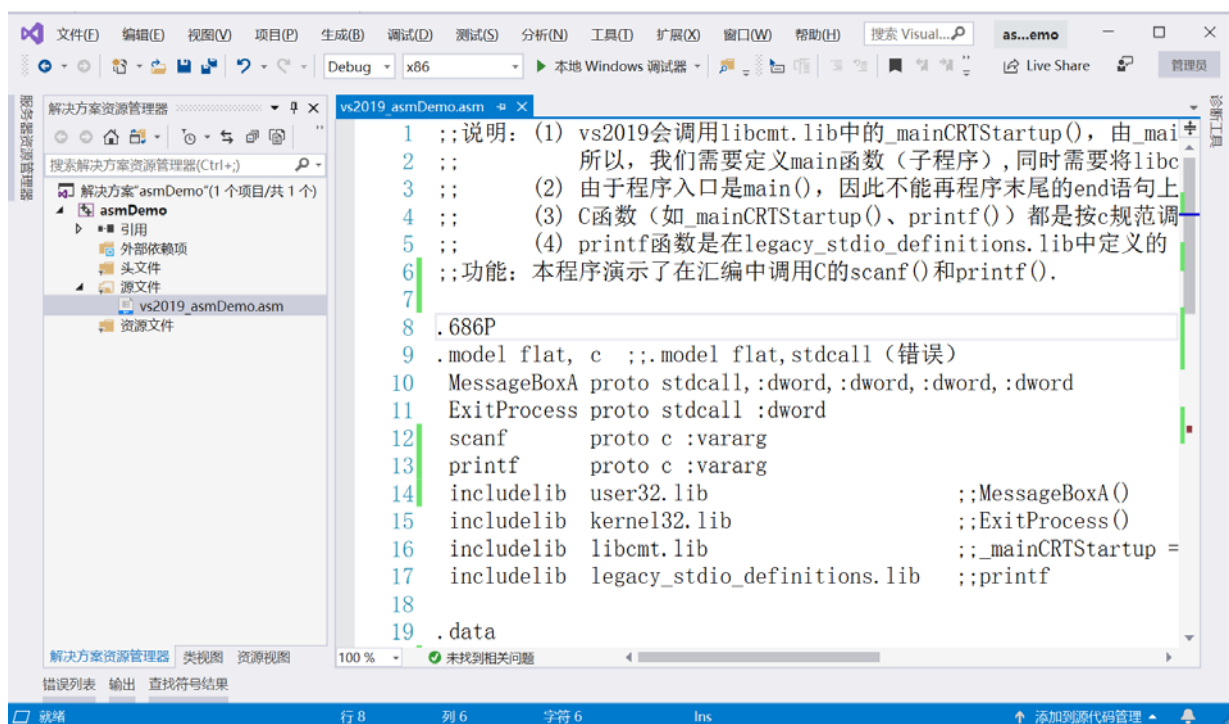
## (2) 增加汇编语言依赖项

在“解决方案资源管理器”窗口鼠标右击工程名“asmDemo” => 生成依赖项 => 生成自定义 => 选择 `masm (.targets, .props)` => 确定;



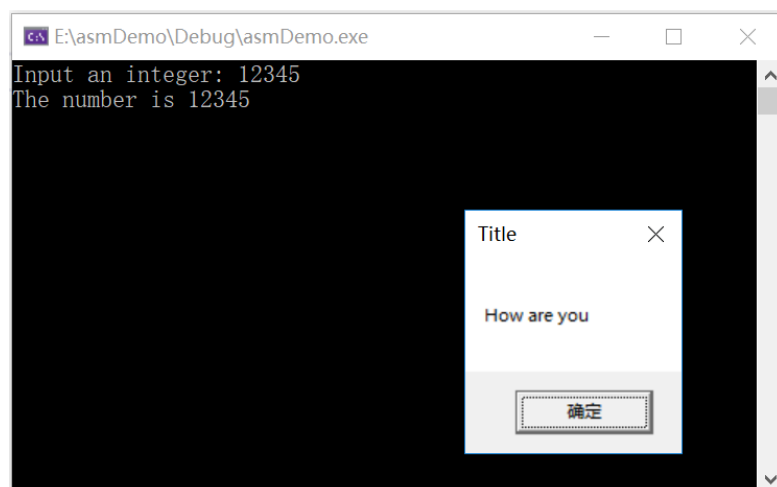
## (3) 导入汇编源文件

菜单“项目” => 添加现有项 => 选择 `vs2019_asmDemo.asm`;



(4) 编译生成 demoAsm.exe 文件: 菜单生成 => 重新生成解决方案;

运行程序: 调试菜单 => 开始执行(不调试), 从键盘输入 12345 后按回车键, 得到程序运行结果:



(5) 也可以利用调试菜单, 进行源码级调试和反汇编。