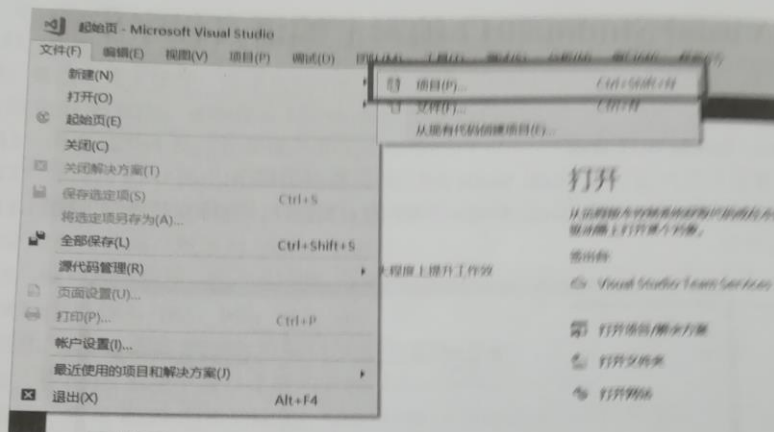
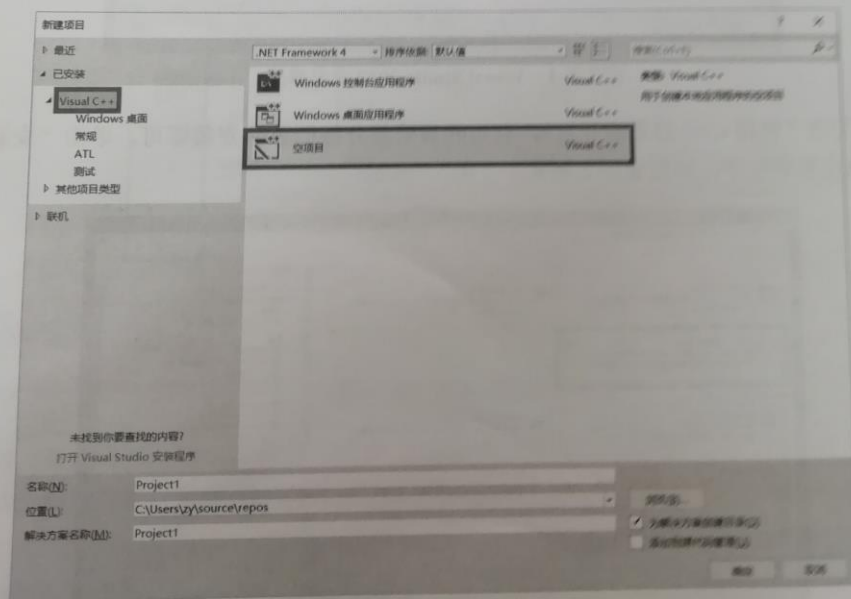


- ④ 安装后重启计算机。
2. 搭建 Visual Studio 2017 汇编语言开发环境
- ① 从官网下载“MASM32 SDK”并安装。
- ② 建立 Visual C++项目。启动 Visual Studio 2017，在菜单栏中选择“文件”→“新建”→“项目”，如附录图 3 所示。



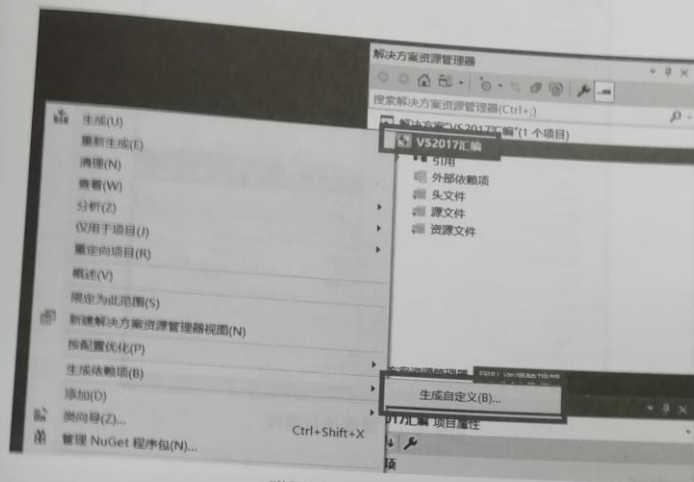
附录图 3 新建 Visual C++项目-1

- ③ 在弹出的对话框中选择“Visual C++”中的“空项目”。填写好项目名称及保存的位置，单击“确定”按钮，如附录图 4 所示。



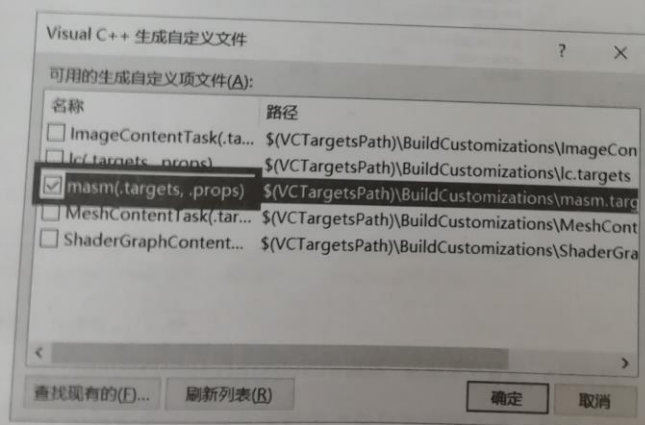
附录图 4 新建 Visual C++项目-2

④ 使用建立好的 Visual C++ 项目配置汇编语言开发环境。在 Visual Studio 2017 窗体右侧的“解决方案管理器”中选中刚刚建立好的项目，右键单击，并选择“生成依赖项”→“生成自定义”，如附录图 5 所示。



附录图 5 生成依赖项

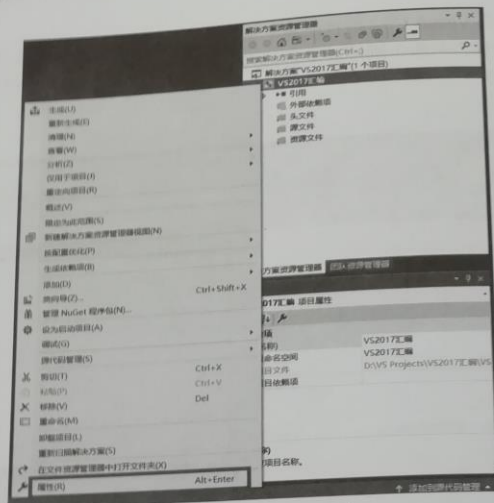
⑤ 在弹出的对话框中勾选“masm(.targets, .props)”，并单击“确定”按钮，如附录图 6 所示。



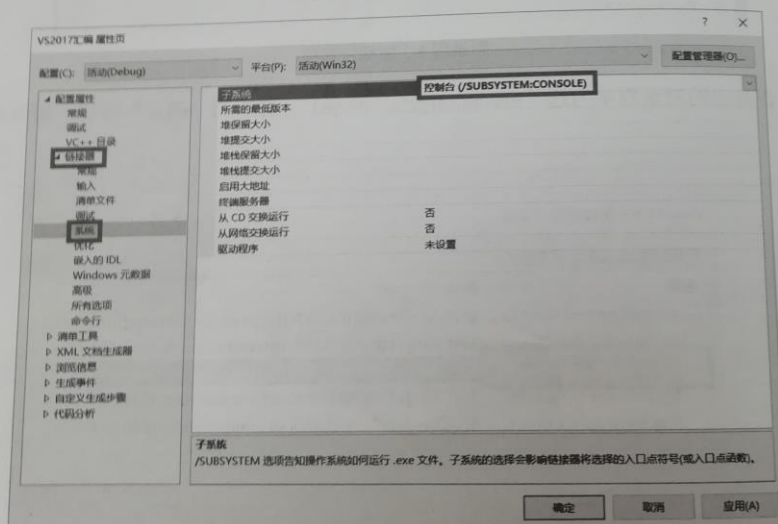
附录图 6 勾选 masm(.targets, .props)

⑥ 在“解决方案管理器”中选中项目，右键单击，并选择“属性”，如附录图 7 所示。

⑦ 在弹出的窗口左侧展开“链接器”节点，选择“系统”选项，将“子系统”选择为“控制台(SUBSYSTEM:CONSOLE)”或“窗口(/SUBSYSTEM:WINDOWS)”，如附录图 8 所示。



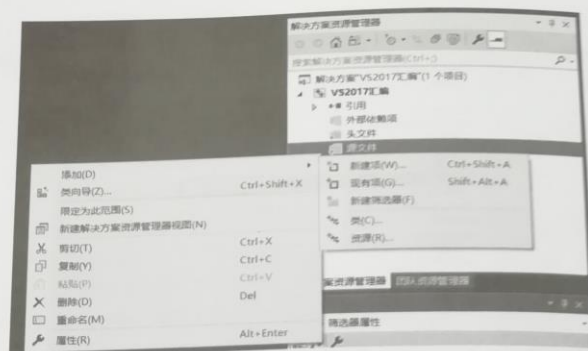
附录图 7 更改项目属性



附录图 8 设置子系统为“控制台(/SUBSYSTEM:CONSOLE)”

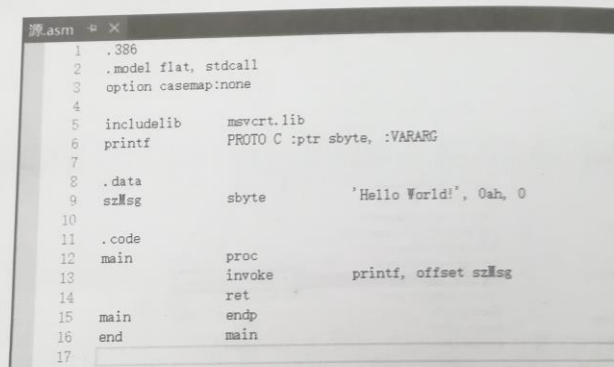
⑧ 选择“常规”选项，在“附加库目录”中填入 masm32 安装目录下的 lib 文件夹路径（例如：C:\masm32\lib），如附录图 9 所示。

⑨ 展开“Microsoft Macro Assembler”节点，选择“General”选项，在“IncludePaths”中填入 masm32 安装目录下的 include 文件夹路径（例如：C:\masm32\include），如附录图 10 所示。



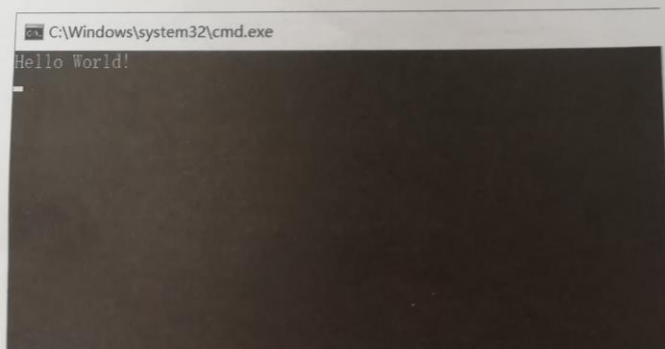
附录图 11 添加汇编语言源文件

已经编写好的汇编代码示例如附录图 12 所示。

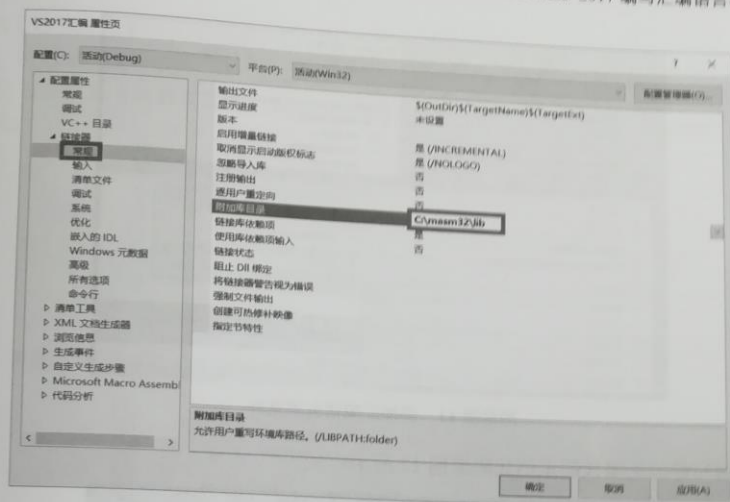


附录图 12 编写好的汇编语言程序代码

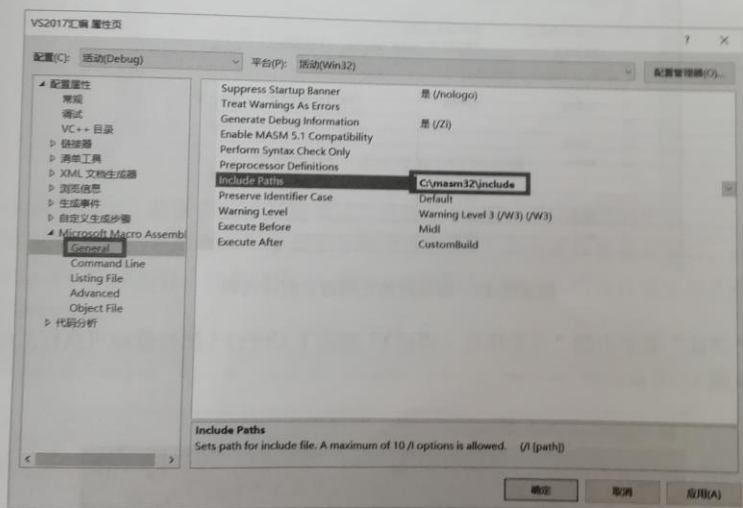
选择“调试”菜单中的“开始执行（调试）”或按下 Ctrl+F5 组合键即可执行汇编语言程序，如附录图 13 所示。



附录图 13 汇编程序执行结果



附录图 9 设置附加库目录



附录图 10 设置 IncludePath

⑩ 添加或新建汇编语言程序。

在“解决方案管理器”中选中项目中的“源文件”并右键单击，选择“添加”→“新建项”（用于编写新的汇编语言程序）或“现有项”（将已有的汇编语言程序添加到当前项目中），如附录图 11 所示。（注：新建项时，需要将文件扩展名设置为.asm）

网上还有适用于 Visual Studio 2017 的汇编语言语法高亮插件, 如有需要, 请自行查找安装。

3. 使用 Visual Studio 2017 调试汇编语言程序

① 首先要将主函数设置为 “main proc...main endp” 的格式, 才能进行调试, 如附录图 14 所示。

```
11 .code
12 main proc
13     invoke     printf, offset szMsg
14     ret
15 main endp
16 end main
```

附录图 14 调试时主程序格式

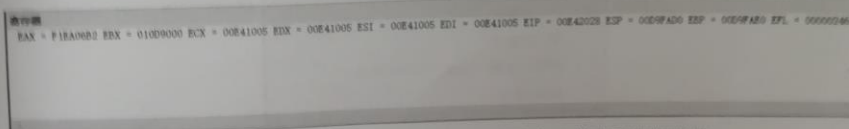
② 鼠标左键单击行号左侧灰色部分设置断点, 如附录图 15 所示。

```
1 .386
2 .model flat, stdcall
3 option casemap:none
4
5 includelib     msvcrt.lib
6 printf         PROTO C :ptr sbyte, :VARARG
7
8 .data
9 szMsg          sbyte     'Hello World!', 0ah, 0
10
11 .code
12 main          proc
13     invoke     printf, offset szMsg
14     ret
15 main          endp
16 end main
```

附录图 15 调试过程中断点设置

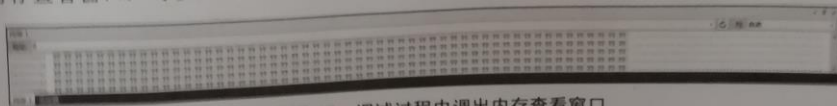
③ 选择 “调试” 菜单下的 “开始调试” 或按下 F5 键, 进入调试的状态。

④ 选择 “调试” → “窗口” 子菜单下的 “寄存器”, 调出寄存器查看窗口, 此时可以看到各个寄存器中存放的值, 如附录图 16 所示。(在调试过程中, 当寄存器值发生改变时, 寄存器值会变成红色)



附录图 16 调试过程中, 调出寄存器查看窗口并查看寄存器值

⑤ 选择 “调试” → “窗口” → “内存” 子菜单下的 “内存 1” ~ “内存 4”, 可以调出 4 个内存查看窗口, 可以同时查看 4 个不同内存区域的内存值, 如附录图 17 所示。



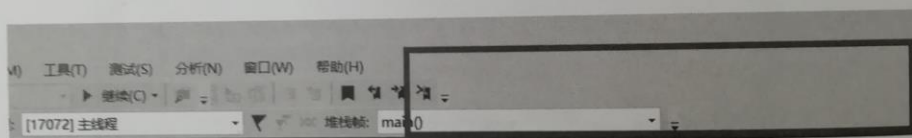
附录图 17 调试过程中调出内存查看窗口

⑥ 内存查看窗口的“地址”一栏可以填写需要查看内存区域的地址。在上述程序“HelloWorld”中，查看“szMsg”变量所处内存区域的值，只需在内存查看窗口的“地址”一栏填写“&szMsg”即可（&是C语言中的取地址符号）。如附录图 18 所示，左侧显示十六进制内容，右侧显示文本内容（在调试过程中，当内存值发生改变时，改变的值会用红色显示）。

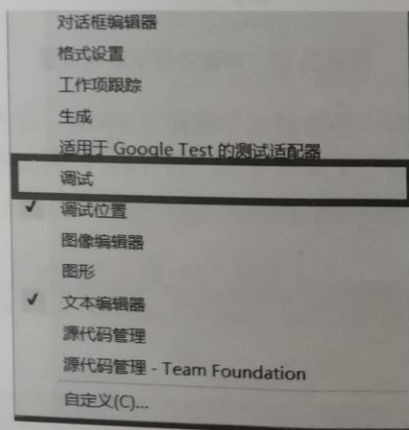
内存 1	
地址: &szMsg	
0x00E45000 48 65 6c 6c 6f 20 57 6f 72 6c 64 21 0a 00 00 00	Hello World!..
0x00E45010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00E45020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00E45030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00E45040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

附录图 18 调试过程中的内存区域值

⑦ 在 Visual Studio 2017 界面上方工具栏空白区域单击右键，勾选“调试”，即可调出调试工具栏，使用“逐语句 (F11)”“逐过程 (F10)”进行汇编语言程序的调试，如附录图 19 (a) 和附录图 19 (b) 所示。



(a)



(b)

附录图 19 调出调试工具栏