

## GCC -S选项：编译非汇编文件

[< 上一页](#)[下一页 >](#)

我们知道，从 C、C++ 源代码生成可执行文件需要经历 4 个过程，分别为预处理、编译、汇编和链接。其中，《[GCC -E选项：对源程序做预处理操作](#)》一节以 demo.c 源程序文件为例，演示了如何使用 GCC 执行预处理操作，并生成了相应的 demo.i 预处理文件。

```
[root@bogon demo]# cat demo.c
#include <stdio.h>
int main(){
    puts("Hello,World!");
    return 0;
}
[root@bogon demo]# gcc -E demo.c -o demo.i
[root@bogon demo]# ls
demo.c demo.i
```

在此基础上，本节继续讲解如何对已得到的预处理文件进行编译，将其再加工为相应的汇编文件。

### gcc -S指令

编译是整个程序构建的核心部分，也是最复杂的部分之一。所谓编译，简单理解就是将预处理得到的程序代码，经过一系列的词法分析、语法分析、语义分析以及优化，加工为当前机器支持的汇编代码。

通过给 gcc 指令添加 -S（注意是大写）选项，即可令 GCC 编译器仅将指定文件加工至编译阶段，并生成对应的汇编代码文件。例如：

```
[root@bogon demo]# gcc -S demo.i
[root@bogon demo]# ls
demo.c demo.i demo.s
```

可以看到，经过执行 gcc -S 指令，其生成了一个名为 demo.s 的文件，这就是经过编译的汇编代码文件。也就是说默认情况下，编译操作会自行新建一个文件名和指定文件相同、后缀名为 .s 的文件，并将编译的结果保存在该文件中。



当然如果需要的话，我们还可以为 gcc -S 指令添加 -o 选项，令 GCC 编译器将编译结果保存在我们指定的文件中。例如：

```
[root@bogon demo]# gcc -S demo.i -o test.i
[root@bogon demo]# ls
demo.c demo.i demo.s test.i
```

需要注意的是，gcc -S 指令操作的文件并非必须是经过预处理后得到的 .i 文件，-S 选项的功能是令 GCC 编译器将指定文件处理至编译阶段结束。这也就意味着，gcc -S 指令可以操作预处理后的 .i 文件，也可以操作源代码文件：

- 如果操作对象为 .i 文件，则 GCC 编译器只需编译此文件；
- 如果操作对象为 .c 或者 .cpp 源代码文件，则 GCC 编译器会对其进行预处理和编译这 2 步操作。

因此，如果我们想直接得到 demo.c 文件对应的汇编文件，就可以借助 gcc -S 指令：

```
[root@bogon demo]# gcc -S demo.c -o demo.s
[root@bogon demo]# ls
demo.c demo.s
```

由此，我们就可以直接获得 demo.c 对应的 demo.s 汇编文件。

对于最终生成的 .s 汇编文件，感兴趣的读者可执行 cat demo.s 指令查看文件中的内容。在此基础上，如果想提高文件内汇编代码的可读性，可以借助 -fverbose-asm 选项，GCC 编译器会自行为汇编代码添加必要的注释，例如：

```
[root@bogon demo]# gcc -S demo.c -fverbose-asm
```

< [上一页](#)

[下一页](#) >

## 所有教程

算法	Python爬虫	C语言入门	C语言编译器	C语言项目案例	数据结构			
多线程	链接库	C++	STL	C++11	socket	GCC	GDB	Makefile
OpenCV	Qt教程	Unity 3D	UE4	游戏引擎	Python	Python并发编程		
TensorFlow	Django	NumPy	Linux	Shell	Java教程	设计模式		
Java Swing	Servlet教程	JSP教程	JSTL	Struts2	Maven	Nexus	↑	