

GCC -c选项：生成目标文件

[< 上一页](#)[下一页 >](#)

我们知道，从 C、C++ 源代码生成可执行文件要经历 4 个过程，分别为预处理、编译、汇编和链接。

《[GCC -E选项：对源程序做预处理操作](#)》一节以 demo.c 源文件为例，介绍了如何对源文件进行预处理，并生成相应的 demo.i 预处理文件；《[GCC -S选项：编译非汇编文件](#)》一节介绍了如何对 demo.c 或者 demo.i 进行编译，并生成相应的 demo.s 汇编代码文件。

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

在此基础上，本节继续讲解如何对已得到的 demo.s 执行汇编操作，并得到相应的目标文件。所谓目标文件，其本质为二进制文件，但由于尚未经过链接操作，所以无法直接运行。

gcc -c指令

简单地理解，汇编其实就是将汇编代码转换成可以执行的机器指令。大部分汇编语句对应一条机器指令，有的汇编语句对应多条机器指令。相对于编译操作，汇编过程会简单很多，它并没有复杂的语法，也没有语义，也不需要做指令优化，只需要根据汇编语句和机器指令的对照表——翻译即可。

通过为 gcc 指令添加 -c 选项（注意是小写字母 c），即可让 GCC 编译器将指定文件加工至汇编阶段，并生成相应的目标文件。例如：

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

可以看到，该指令生成了和 demo.s 同名但后缀名为 .o 的文件，这就是经过汇编操作得到的目标文件。

当然如果必要的话，还可以为 gcc -c 指令在添加一个 -o 选项，用于将汇编操作的结果输入到指定文件中，例如：

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

需要强调的一点是，和 gcc -S 类似，gcc -c 选项并非只能用于加工 .s 文件。事实上，-c 选项只是令 GCC 编译器将指定文件加工至汇编阶段，但不执行链接操作。这也就意味着：

- 如果指定文件为源程序文件（例如 demo.c），则 gcc -c 指令会对 demo.c 文件执行预处理、编译以及汇编这 3 步操作；
- 如果指定文件为刚刚经过预处理后的文件（例如 demo.i），则 gcc -c 指令对 demo.i 文件执行编译和汇编这 2 步操作；
- 如果指定文件为刚刚经过编译后的文件（例如 demo.s），则 gcc -c 指令只对 demo.s 文件执行汇编这 1 步操作。

注意，如果指定文件已经经过汇编，或者 GCC 编译器无法识别，则 gcc -c 指令不做任何操作。

这里以 demo.c、demo.i、demo.s 为例，演示 gcc -c 指令的作用：

```
[root@bogon demo]# ls
demo.c demo.i demo.s
[root@bogon demo]# gcc -c demo.c -o democ.o
[root@bogon demo]# ls
demo.c demo.i democ.o demo.s
[root@bogon demo]# gcc -c demo.i -o demoi.o
[root@bogon demo]# ls
demo.c demo.i demoi.o democ.o demo.s
[root@bogon demo]# gcc -c demo.s -o demos.o
[root@bogon demo]# ls
demo.c demo.i demoi.o democ.o demo.s demos.o
```



以上操作分别生成的 democ.o、demoi.o 以及 demos.o，其包含的二进制内容是完全一样的，只是文件名不同而已。

gcc执行链接操作

得到生成目标文件之后，接下来就可以直接使用 gcc 指令继续执行链接操作，例如：

```
[root@bogon demo]# gcc democ.o -o democ.exe
[root@bogon demo]# ./democ.exe
Hello,World!
[root@bogon demo]# gcc demoi.o -o demoi.exe
[root@bogon demo]# ./demoi.exe
Hello,World!
[root@bogon demo]# gcc demos.o -o demos.exe
[root@bogon demo]# ./demos.exe
Hello,World!
```

gcc 会根据所给文件的后缀名 .o，自行判断出此类文件为目标文件，仅需要进行链接操作，所以这里的 gcc 指令只会对 democ.o、demoi.o、demos.i 执行链接操作，并分别生成 democ.exe、demoi.exe 以及 demos.exe 这 3 个可执行文件。

通过分别执行这 3 个可执行文件，其执行结果完全相同，从侧面验证了 democ.o、demoi.o、demos.o 包含二进制的內容相同。

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

Spring

Spring MVC

Spring Boot

Spring Cloud

Hibernate

Mybatis

MySQL教程

MySQL函数

NoSQL

Redis常用命令手册

HBase

MongoDB

Go语言

C#

MATLAB

JavaScript

Bootstrap

HTML

CSS

PHP

汇编语言

TCP/IP

vi命令

Android教程

区块链

Docker

大数据

↑