

## GCC -E选项：对源程序做预处理操作

< [上一页](#)[下一页](#) >

通过前面的学习我们知道，根据源代码文件后缀名的不同，gcc 指令可以自行判断出该源程序是由哪种编程语言编写，从而完成由源代码到可执行代码的转换工作。考虑到本教程主要讲解如何使用 gcc 指令运行 C/C++ 程序，因此本节就以 C 语言程序为例（如下所示），讲解 gcc -E 选项的功能和用法。

```
01. //存储在 demo.c 文件中
02. #include <stdio.h>
03. int main() {
04.     puts("GCC教程: http://c.biancheng.net/gcc/");
05.     return 0;
06. }
```

无论是 C 还是 C++ 程序，其从源代码转变为可执行代码的过程，具体可分为 4 个过程，分别为预处理（Preprocessing）、编译（Compilation）、汇编（Assembly）和链接（Linking）。默认情况下，gcc 指令会一气呵成，直接将源代码历经这 4 个过程转变为可执行代码，且不会保留各个阶段产生的中间文件（可阅读《[GCC编译C/C++程序（一步完成）](#)》一节）。

而如果想查看这 4 个阶段各自产生的中间文件，最简单直接的方式就是对源代码进行“分步编译”，即控制 GCC 编译器逐步对源代码进行预处理、编译、汇编以及链接操作。其中，通过为 gcc 指令添加 -E 选项，即可控制 GCC 编译器仅对源代码做预处理操作。

所谓预处理操作，主要是处理那些源文件和头文件中以 # 开头的命令（比如 #include、#define、#ifdef 等），并删除程序中所有的注释 // 和 /\* ... \*/。有关预处理操作的具体细节，读者可阅读《[那些被编译器隐藏了的过程](#)》一节做详细了解。

值得注意的是，默认情况下 gcc -E 指令只会将预处理操作的结果输出到屏幕上，并不会自动保存到某个文件。因此该指令往往会和 -o 选项连用，将结果导入到指令的文件中。比如：

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

读者可自行去掉 -o demo.i 部分，查看该指令的执行结果。



Linux 系统中通常用 ".i" 作为 C 语言程序预处理后所得文件的后缀名。由此，就完成了 demo.c 文件的预处理操作，并将其结果导入到了 demo.i 文件中。

读者可以通过执行 `cat demo.i` 指令查看该文件中的内容，但通常没有足够 C 语言功底的患者是看不懂的。为此，我们可以为 gcc 指令再添加一个 -C 选项，阻止 GCC 删除源文件和头文件中的注释：

```
[root@bogon demo]# gcc -E -C demo.c -o demo.i
```

注意，这里是大写的 -C，不是小写的 -c。小写的 -c 另作他用，后续章节会做详细讲解。

## gcc -E支持的常用选项

除了 -C、-o 以外，根据实际场景的需要，gcc -E 后面还可以添加其它的选项，例如：

表 1 gcc -E 常用选项

| 选 项                            | 功 能  |
|--------------------------------|--|
| -D<br>name[=definition]        | 在处理源文件之前，先定义宏 name。宏 name 必须是在源文件和头文件中都没有被定义过的。将该选项搭配源代码中的 #ifdef name 命令使用，可以实现条件式编译。如果没有指定一个替换的值（即省略 =definition），该宏被定义为值 1。 |
| -U name                        | 如果在命令行或 GCC 默认设置中定义过宏 name，则“取消”name 的定义。-D 和 -U 选项会依据在命令行中出现的先后顺序进行处理。  |
| -include file                  | 如同在源代码中添加 #include "file" 一样。  |
| -iquote dir                    | 对于以引号（#include ""）导入的头文件中，-iquote 指令可以指定该头文件的搜索路径。当 GCC 在源程序所在目录下找不到此头文件时，就会去 -iquote 指令指定的目录中查找。                              |
| -I dir                         | 同时适用于以引号 "" 和 <> 导入的头文件。当 GCC 在 -iquote 指令指定的目录下搜索头文件失败时，会再自动去 -I 指定的目录中查找。该选项在 GCC 10.1 版本中已被弃用，并建议用 -iquote 选项代替。            |
| -isystem dir<br>-idirafter dir | 都用于指定搜索头文件的目录，适用于以引号 "" 和 <> 导入的头文件。   |

其中，对于指定 #include 搜索路径的几个选项，作用的先后顺序如下：

- 对于用 #include "" 引号形式引入的头文件，首先搜索当前程序文件所在的目录，其次再前往 -iquote 选项指定的目录中查找；
- 前往 -I 选项指定的目录中搜索；
- 前往 -isystem 选项指定的目录中搜索；



- 前往默认的系统路径下搜索；
- 前往 -idirafter 选项指定的目录中搜索。

除表 1 罗列的几个选项之外，预处理过程可以使用的选项还有很多，比如 -imacros、-undef、-M 等，感兴趣的读者可前往官网 [GCC10.1.0 预处理过程选项查看](#)。

< [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

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

云计算

推荐阅读

编程笔记

资源下载

VIP视频

一对一答疑

关于我们

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2020 biancheng.net, 陕ICP备15000209号

biancheng.net

